THE MICRO CHECK-IN FRAMEWORK FOR THE Next GENERATION OF LOCATION-BASED APPLICATIONS

A Thesis

Presented to

the Faculty of California Polytechnic State University

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Shraddha Kacha

June 2012
COMMITTEE MEMBERSHIP

TITLE: The Micro Check-in Framework for the Next Generation of Location-based Applications

AUTHOR: Shraddha Kacha

DATE SUBMITTED: June 2012

COMMITTEE CO-CHAIR: Dr. Franz Kurfess

COMMITTEE CO-CHAIR: Dr. John Bellardo

COMMITTEE MEMBER: Dr. Gene Fisher
To my parents,

for their support and encouragement.

If I had been with them, as I wished, and they deserved,

this thesis would not have been completed.

And to my loving husband, Kaushal Solanki,

whose energy inspired me countless number of times. He has been the biggest source of inspiration for me throughout the year. Without his help, support and motivation it would have been impossible to finish my thesis.
Abstract

The Micro Check-in Framework for the Next Generation of Location-based Applications

Shraddha Kacha

Today’s mobile computing platforms provide new, convenient possibilities for location-based commercial activity. In particular, the global positioning system (GPS) provides location data for local advertising and retail activity. In this thesis, we take the next step and introduce the concept of micro check-in. Micro check-in is a simple framework that allows users to match their current location to specific objects or places at a resolution much finer than that allowed by GPS based check-in. We show that micro check-in enables customers to reveal their location at a fine level of granularity. By describing the concepts of points of service, service providers, users, and the enabling technology, we present a generalized, abstract, framework for designing, developing, and analyzing micro check-in based systems.

We discuss several new applications that open up as a result of the combination of real-life and digital communication channel as a result of the micro check-in process. For example, micro check-in could prove quite useful to customers at a restaurant table, a gym equipment station, an exhibit at a museum and many more. This thesis studies micro check-in applied to the process of ordering food from a table at a restaurant and present the beneficial tradeoffs enabled by micro check-in. We find that the micro check-in framework has promising applications in several areas, but specifically utilize the restaurant application, OnTable, to prove the potential use of the micro check-in framework.
Acknowledgements

I would like to express sincere gratitude to my committee members, Professors, John Bellardo, Franz Kurfess and Gene Fisher, for their guidance, encouragement, and support. I am extremely fortunate to have the opportunity to work closely with three outstanding, highly knowledgeable, yet very friendly persons. Professors Kurfess and Bellardo have always been very patient and understanding with me. I would like to thank both for their insightful comments and suggestions throughout my graduate studies, which have greatly improved the clarity of this thesis.

I would like to thank Professor Bellardo for inspiring my passion and putting in trust on me to pursue my thesis idea. Without his support and instinct this thesis would not have been started. He helped me a lot to think in right direction and take smart decisions. Professors Bellardo has always been available for detailed technical discussions, which has shaped my thinking through the year.

I would like to thank Professor Kurfess for allowing me to conduct usability survey in his classes. This survey is the biggest validation factor in my thesis. Without his help the validation for the thesis would have been impossible. In addition, his usability expertise helped me to focus on creating an application that is usable by people. Professor Kurfess has always been supportive to help me move in right direction.

Professor Fisher has always helped me to think from the perspective of a software engineer and encouraged me to fulfill my goals as a graduate student.

I would like to thank Professor Clark Turner for his help and support in writing my thesis, inspiring me to participate in Innovation Quest and encouraging me to become a successful software engineer in practice. He has been instrumen-
tal in my development as a student; my writing skills have improved manyfold under his tutelage.

I would like to thank Professor Alex Dekhtyar, who motivated me to think beyond the horizon and helped me to focus in right direction. His support and enthusiasm kept me motivated throughout the year.

It was pleasure working with Sandra Ngo, who always worked hard to achieve our goals in developing a mobile application throughout the year. I would like to thank Kelly Watkins for her awesome user interface designs while she was involved in the project. I am thankful to Carsten Pfeffer for his help, at various stages, in the development of web application.

I have no words to express gratitude towards my family, without whose support and encouragement, this journey would not even have begun. I especially thank my husband, Kaushal Solanki, for every bit of his help, support, and encouragement. He made sure that I do not have to worry about anything else while I was working on the thesis, or for an approaching deadline. Special thanks to my close friends, Ruchi, and Sapna for their continuous love, friendship, and the fun we had. You both have always been with me in my good and bad times alike. I am very grateful to my loving sisters Jagruti and Neeta for their unconditional love, help, support and encouragement. I would also thank my sister-in-law Dhara, her husband Sachin, my cousins Nainesh, Krunal and Jaydip for their immense help and support throughout my graduate studies. My parents deserve all the credit for what I have achieved, and what I may, in the future. To them, nothing was more important than my education. They asked for more when I was happy with what I had, and inspired and supported me when I was not. I am grateful to my late grandmother and grandfather for the endless love and blessings.
# Contents

List of Tables \hfill \pageref{sec:loft}  
List of Figures \hfill \pageref{sec:loff}  

1 Introduction \hfill 1  

2 Background \hfill 4  
\hspace{1em} 2.1 Smartphone Platforms \hfill 5  
\hspace{1em} 2.2 Revealing Location \hfill 6  
\hspace{1em} 2.3 Technology Adoption at Restaurants \hfill 8  
\hspace{1em} 2.4 Summary \hfill 9  

3 Micro Check-in and its Applications \hfill 10  
\hspace{1em} 3.1 What is Micro Check-in? \hfill 10  
\hspace{1em} 3.2 Why Micro Check-in? \hfill 12  
\hspace{1em} 3.3 How to Micro Check-in? \hfill 12  
\hspace{1em} \hspace{1em} 3.3.1 Enabling Technologies \hfill 14  
\hspace{1em} \hspace{1em} 3.3.2 The Setup \hfill 16  
\hspace{1em} \hspace{1em} 3.3.3 The Workflow \hfill 19  
\hspace{1em} 3.4 Micro Check-in Applications \hfill 21  
\hspace{1em} \hspace{1em} 3.4.1 Example Applications \hfill 22  

4 OnTable: Design and Engineering \hfill 28  
\hspace{1em} 4.1 What is OnTable? \hfill 29  
\hspace{1em} 4.2 Business Analysis \hfill 29  
\hspace{1em} 4.3 The Architecture, Setup and Workflow \hfill 31  
\hspace{1em} \hspace{1em} 4.3.1 Mobile Application \hfill 35  

4.3.2 Web Application ........................................... 35
4.4 Wire Frames for OnTable Application .................... 36

5 **Implementation and Results** 42
5.1 Implementation Details ................................. 42
  5.1.1 Web Application for Restaurants ................. 43
  5.1.2 Smartphone Application for Users ............ 46
5.2 Testing .................................................. 48
  5.2.1 Testing framework for Web Server (back-end) .... 49
  5.2.2 Testing Framework for Smartphone Application ... 50
  5.2.3 Performance Testing for Web Server ............ 52
5.3 Results ................................................. 55
  5.3.1 Concept validation ................................. 57
  5.3.2 User acceptance validation ....................... 58
  5.3.3 Efficiency validation .............................. 59
  5.3.4 Fewer errors validation ......................... 61
5.4 Summary ............................................... 62

6 **Validation** 64
6.1 Beta Release Goals .................................... 64
6.2 Data Gathering Framework ............................ 65
6.3 Metrics ................................................. 66
6.4 Threats to Validity ................................... 67

7 **Future Work and Conclusions** 69
7.1 Future Work ........................................... 69
7.2 Conclusions ............................................ 70

**Bibliography** 71
List of Tables

2.1 Check-in service users by smartphone platform. The data is obtained from a comScore study [2]. Index = % of Check-In Service Users / % of Smartphone Users x 100 ........................................ 7

2.2 Demographic profile for check-in service users. The data is obtained from a comScore study [2]. .......................................... 7

6.1 Validation metrics and how we measure it. .......................... 66
List of Figures

3.1 Currently available technologies that can enable micro check-in.  . 13
3.2 Setup of a system that allows users to reveal location at specific
service points to receive services from a service provider. . . . . . 17
3.3 Functional block diagram of a general micro check-in based system. 18
3.4 Flowchart that describes the workflow for the user requesting ser-
vice using a mobile device. . . . . . . . . . . . . . . . . . . . . . 20
3.5 Flowchart describing the workflow of how a user reveals location
at high resolution (“micro check-in”) to receive service(s) from a
provider at a point of service. . . . . . . . . . . . . . . . . . . . . 21
3.6 Flowchart that describes the workflow for the service provider. . 22

4.1 High level architecture of OnTable. . . . . . . . . . . . . . . . . 32
4.2 The setup for OnTable, a micro check-in application for restaurant
ordering. Notice how simple adaptation of various blocks of Figure
3.2 allows description of this new micro check-in based application. 33
4.3 Functional block diagram of OnTable. Notice how the simple adap-
tation of various blocks of Figure 3.3 allows description of this new
micro check-in based application. . . . . . . . . . . . . . . . . . 34
4.4 Mobile application flow control diagram. . . . . . . . . . . . . 36
4.5 Web application flow control diagram. . . . . . . . . . . . . . 37
4.6 A user can browse the selected restaurant’s menu on a smartphone.
User can also look at the different categories of food in the menu.
User also gets a chance to add items to the tray. . . . . . . . . . 38
4.7 Micro check-in to a table: Once a user decides to place an order, he/she must scan the QR code placed on the table. The smartphone’s GPS should be enabled for additional security. A web server will verify the order request based on the user’s QR code data and the location data. ................................. 39

4.8 Favorites: The user can add a dish to his/her favorites. He/she can add notes and take a picture of the favorite-ed dish. He/she can also share it with friends on Facebook and/or Twitter. .................. 40

4.9 Once micro checked-in to a restaurant, a user can place an order from his/her tray. He/she also has the option to add the selected items to favorites. ................................................................. 41

5.1 Home page of the web application showing tables and order information for each tables. ................................................................. 46

5.2 A popup window that shows the order information with item name, number of items and notes for a selected table. ......................... 47

5.3 The home page of the application showing example near by restaurants based on user’s current GPS location ......................... 48

5.4 An example page while browsing a menu of a selected restaurant through different categories. ................................................................. 49

5.5 Edit item page that allows users to write notes and change the quantity of the item. ................................................................. 50

5.6 A page showing a tray for a selected restaurant. ......................... 51

5.7 Order polling time interval(msec) versus delay (sec) in logarithmic scale. ................................................................. 53

5.8 Server load(req/sec) versus delay in(sec). ................................................................. 54

5.9 A sample QR code used for the usability survey. ......................... 56

5.10 Results for the question: “In your eating-out experience at a sit down restaurant, what are some problems and/or frustrations that you typically encounter?” The Y-axis represents number of people. People may select more than one checkbox, so percentages may add up to more than 100%. ................................................................. 58

5.11 Results for the question: “How likely would you use this application if it is offered by a restaurant for ordering food from the table using your smartphone?” The Y-axis represents number of people. ................................................................. 59

5.12 Results for the question: “What would be more efficient to order food while sitting at a restaurant table?” The Y-axis represents number of people. ................................................................. 60
5.13 Results for the question: “OnTable helps to eliminate possible errors by placing an order accurately using smartphone. How much do you agree with the statement?” The Y-axis represents number of people.
Chapter 1

Introduction

The smartphone is the latest in a series of technological innovations that have arguably fundamentally altered our lives for the better. Smartphones provide many unprecedented capabilities right at the user’s fingertips. That has been made possible due to the convergence of functionalities such as fast computing chips, vibrant touch screens, high-resolution cameras, music players, email clients, full-function web browsers, global positioning system (GPS) receivers, social networking software, and so on. The smartphone is rapidly becoming the primary communication and entertainment device [3].

This mass adoption of smartphones coupled with people’s understanding and acceptance of new smartphone-based technologies, provides a unique opportunity for innovations that can greatly enhance user experiences. The software applications for the smartphone platforms that we are seeing today just indicate the beginning of this innovative trend. Technology visionaries are already forecasting that many if not most of the special-purpose objects around us are going to be replaced by apps running on tablets or smartphones [8].
This thesis proposes the concept of micro check-in as a framework that can positively influence many real-life interactions. The micro check-in framework allows users to reveal their location (“check-in”) at a much finer resolution than is possible by technologies such as GPS. It is a high resolution check-in at a specific object rather than a place. Service providers can define specific points of service, defined as objects or entities that need a higher degree of precision to localize (e.g., a museum exhibit, a restaurant table, or a classroom desk). Users can micro check-in to these points of service to reveal their location to the service providers, and in return receive an enhanced experience and/or a more efficient service. There are several ways of enabling micro check-in that include image recognition, near field communication (NFC), and quick response codes (QR codes), as discussed in Chapter 3.

The key contribution of this thesis is the detailed presentation of an abstract, generalized, framework that can form the basis of new applications in various fields. One piece of evidence that this abstraction is working is that whenever a person is presented with the basic idea of micro check-in, he or she starts thinking and usually comes up with entirely new application scenarios. With this, we envision that the applications based on micro check-in framework will make mobile phones much more useful by improving and enhancing people’s interaction with the real world. A few applications already exist that provide functionalities that resemble the micro check-in scenario.

To illustrate the practical real-life use of the micro check-in framework, we discuss several potential applications in Chapter 3. To demonstrate how easily these applications can be designed from the general framework, this thesis specifically studies and investigates one application, that of ordering food by micro check-in at a restaurant table. The software, named OnTable, is a collection of
mobile and web frontends, and a web server, that allows users to scan a QR code to micro check-in at a table and place an order. The order goes to kitchen in real-time along with the ID of the table it came from. Once the order is prepared, the wait staff will serve the food at the user’s table. We believe that OnTable will help restaurants by providing increased table turnarounds, less queue time, fewer errors and better service. Similar benefits can often be realized for other applications as well.

Throughout this thesis, I refer to Dr. Bellardo, Dr.Kurfess, and myself as we. We start by providing the necessary background in the next chapter to motivate the use of micro check-in framework and the presented restaurant ordering application.
Chapter 2

Background

A smartphone is a high-end phone built on a mobile computing platform, with more advanced computing ability and connectivity than a contemporary feature phone. Smartphones are redefining a number of our day-to-day activities. With the emergence of this highly portable device several capabilities including full-motion cameras, bright and large high-density screens, music player, email client, Global Positioning System (GPS), Wi-Fi, Third Generation and Fourth Generation (3G/4G) data connectivity, and innovative new applications have become available, and are collectively changing our lives for the better.

Of these, social location check-in applications are becoming quite popular (check popular applications such as Foursquare [6], and Loopt [15]). People reveal their location based on GPS receivers for several purposes, including connecting with friends, earning rewards and badges, and getting special promotions. In this thesis, we introduce the concept of micro check-in, wherein users selectively reveal their location at a very fine resolution, such as at a restaurant table or at a museum exhibit. This finer level of check-in detail allows several applications that are not possible otherwise. This thesis takes a methodological way of exam-
ining the advantages and drawbacks of our approach in comparison with existing solutions to similar problems.

In this chapter, we present the background to motivate the idea of micro check-in and its applications. In Section 2.1, we discuss the smartphone platforms and how its widespread adoption is changing the consumer application space. Here we also go over the popular platforms and their market share. We also present some background about the check-in applications available today in Section 2.2. As mentioned in Chapter 1, we are delving deeper into a specific application, that of creating a real-time system that allows patrons to order food by micro checking-in to a table. Hence, we provide a background of the restaurant industry and their openness to adopt technology in Section 2.3. We now start our background discussion with smartphone platforms in the following section.

2.1 Smartphone Platforms

Today, mobile phones are far more than a portable device to have a voice conversation with a remote individual. It is widely accepted among popular media as well as technology visionaries that the emergence of a smartphone represents a new era of personalization of computing. The number of smartphones shipped recently surpassed the number of shipped personal computers [18]. This represents a strong trend that is going to continue and has the potential to fundamentally change our interaction with the digital as well as the real world.

The most popular smartphone platforms include Google Android OS, Apple iOS, Microsoft Windows phone, Symbian and Research in Motion’s (RIM) BlackBerry. As of today, Android OS is the top smartphone platform worldwide with 34.7 percent market share. Apple is in the second position, accounting for 25.5
percent of the smartphone market. RIM ranked third with 27.1 percent share, down 4.6 percentage points from the preceding period, followed by Microsoft (7.5 percent) and Symbian (2.3 percent). The statistics are shown in Table 2.1. Having discussed the smartphone platforms, we now move on to the specific application of revealing location, or check-in, in the following section.

2.2 Revealing Location

One of the most popular application of smartphones is social check-in applications. Check-in simply means revealing your location [2]. There are social networking services, such as Foursquare [6], Google Latitude [14], Facebook [5], and Gowalla [7] (recently acquired by Facebook [24]) which allow users to check in to a physical place and share their location with their friends or general public. Users can check in to a specific location using an application on a smartphone. The application will use the phone’s GPS to find their current location.

A recent survey shows that nearly one in five smart phone owners access check-in services via their mobile device, as described in a ComScore article\(^1\) and summarized in Table 2.1. It can be seen that, a significant percentage of users of the two most popular platforms, iOS and Android, use check-in services. This shows that users are more and more aware of the potential benefits of revealing location. The demographic profile of users of check-in services is summarized in Table 2.2. With this, we believe that people are ready to accept, understand, and use a finer level of check-in, as studied in this thesis.

The ability to interact with consumers on this local level through special offers, deals and other provides brands with the real-time opportunity to en-

---

\(^1\)ComScore, Inc. (NASDAQ: SCOR), is a leader in measuring the digital world.
gage consumers through their mobile device. Among all these local businesses, restaurants represent a large market with clear need to engage with customers and provide better service to them via their smartphones. A smartphone based framework for restaurants is a motivating application. In the following section, we discuss this opportunity with the restaurant industry.

Table 2.1: Check-in service users by smartphone platform. The data is obtained from a comScore study [2]. Index = % of Check-In Service Users / % of Smartphone Users x 100

<table>
<thead>
<tr>
<th>% of Smartphone Users</th>
<th>% of Check-In Service Users</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Smartphone Audience: 13+ yrs old</td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>Android</td>
<td>34.7%</td>
<td>36.6%</td>
</tr>
<tr>
<td>Apple</td>
<td>25.5%</td>
<td>33.7%</td>
</tr>
<tr>
<td>Microsoft</td>
<td>7.5%</td>
<td>4.3%</td>
</tr>
<tr>
<td>Palm</td>
<td>2.8%</td>
<td>2.0%</td>
</tr>
<tr>
<td>Symbian</td>
<td>2.3%</td>
<td>1.2%</td>
</tr>
</tbody>
</table>

Table 2.2: Demographic profile for check-in service users. The data is obtained from a comScore study [2].

<table>
<thead>
<tr>
<th>% of Total Mobile Users</th>
<th>% of Smartphone Users</th>
<th>% of Check-In Service Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Audience: 13+ yrs old</td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>Male</td>
<td>48.0%</td>
<td>53.9%</td>
</tr>
<tr>
<td>Female</td>
<td>52.0%</td>
<td>46.1%</td>
</tr>
<tr>
<td>Age: 13-17</td>
<td>7.1%</td>
<td>6.0%</td>
</tr>
<tr>
<td>Age: 18-24</td>
<td>12.5%</td>
<td>17.5%</td>
</tr>
<tr>
<td>Age: 25-34</td>
<td>17.6%</td>
<td>27.3%</td>
</tr>
<tr>
<td>Age: 35-44</td>
<td>16.8%</td>
<td>21.8%</td>
</tr>
<tr>
<td>Age: 45-54</td>
<td>17.8%</td>
<td>15.5%</td>
</tr>
</tbody>
</table>
2.3 Technology Adoption at Restaurants

A restaurant is a place where people prepare and serve food, drinks, and desserts to patrons in exchange for money [17]. There are many types of restaurants including formal, semi formal, casual and semi casual, fast food, lunch spots, hotel restaurants, breakfast diners, pizza joints and take out. Full-service restaurants provide the traditional dine-in experience to the patrons. In these full-service restaurants, a server takes the order and passes it on to the kitchen. The food is then prepared and served to the customer.

The concept of restaurants is centuries old and has a rich history. They have greatly evolved over the years, and technology has played an important role in how this evolution has taken shape. Over the past decades, the use of computers and customized software to manage customers and inventory has greatly simplified and streamlined restaurant operations. Point of sales (POS) software helps restaurant staff enter and process orders, and has become a nearly-essential component of restaurant management. With the advent and popularity of the Internet, several restaurant and food-related applications have become available in the market, including restaurant reservation, online food ordering [9], restaurant reviewing/rating [27], and locating restaurants using GPS [14].

The restaurant industry is now opening up to adopt new technologies to make their customer’s experience better. This is also influenced by changing expectations of the customers who have grown more and more accustomed to using their smartphones for variety of tasks. The traditional song and dance experience where the customer takes their seats, waits, orders food/drinks, again wait for food to arrive, eats food, and finally waits to get their check, may be frustrating, particularly during peak hours. This is the problem that we address
using the proposed application, named **OnTable** (see Chapter 4 for details). In the following section, we provide a brief summary of the background discussed in this chapter.

### 2.4 Summary

In summary, the following are the critical observations related to the state of the consumer and restaurant industry today, which are particularly relevant to the proposed work.

- Fast growing smartphone use, and users who embrace the use of advanced technology.

- Several new applications have opened up due to convergence of fast processors, touch-screens, powerful cameras, GPS, and other capabilities. Revealing location via check-in is one such example.

- People are open and positive about doing more and more via their smartphones.

- The restaurant industry is growing and is now much more open to adoption of technology, as a means of attracting more patrons.
Chapter 3

Micro Check-in and its Applications

We introduce the concept of micro check-in in this chapter, which is one of the central ideas proposed in this thesis. The concept is introduced and explained in Section 3.1. Next, in Section 3.2, we discuss the benefits to the user and to the businesses that enable micro check-in for its users. In Section 3.3, we suggest several technologies that can be employed to allow this finer level of check-in. These include image recognition, near-field communication (NFC), and quick-response codes (QR codes). As discussed in Chapter 1, we analyze, design and implement an application for the restaurant market, that is enabled by micro check-in at a restaurant table.

3.1 What is Micro Check-in?

Micro check-in is a framework that allows users to check-in to specific objects or places at a resolution much finer than that allowed by GPS or location based
check-in. For example, when location information is available based on GPS, one can deduce the location at the general area, such as a restaurant, park, or a mall but not a precise place/location. This is the limit on the accuracy of a GPS device today. This also limits the potential applications of such systems to cases such as social networking, and earning badges and rewards. There are many more possibilities when users could selectively reveal their location to specific entities at a much finer resolution. Examples include checking in at a gym machine, a painting, an instruction board, a classroom desk, a museum exhibit, an airplane seat, an item for sale at a supermarket, a hotel room, and steering wheel of a car.

Following are the main advantages of micro check-in over currently popular GPS based check-in.

- **Greater accuracy and precision**: Micro check-in happens by a deliberate action of the user (e.g., scanning a QR code), and hence it is both more accurate (*high resolution*) and precise (*highly repeatable*) in determining the location.

- **Defines a specific service point**: By strategically placing the check-in points, one can define the specific service points that are relevant to the user as well as the service provider. This is an important point to note: while micro check-in defines higher resolution of location, it does not allow check-in *anywhere*. This check-in must happen at a pre-defined service point. A restaurant table or a gym machine are two obvious examples, also discussed later.

- **New applications**: We believe that several new applications will be enabled by micro check-in. These include new efficient ways of providing existing services, as well as entirely new possibilities not yet available. We
discuss this more in the following section.

3.2 Why Micro Check-in?

There are numerous cases where your specific location is crucial for a particular benefit or service. Below we list a few examples to illustrate the idea that several use cases require the knowledge of the precise location of the user.

- To serve you, a full-service restaurant needs to know the exact table where you are.

- For the flight attendant to come to your seat to fulfill your request, he or she needs to know the seat number where you are seated.

There is a huge potential for technical and business innovations around the micro check-in. Some of the above are improved ways of providing existing service (e.g., restaurant and flight seat check-in), while some others are entirely new possibilities (such as viewing a wine-related video by checking-in on a specific wine).

3.3 How to Micro Check-in?

The way in which micro check-in is practically realized is an important aspect of the overall benefit of micro check-in to the service provider and the user. There is more than one way to enable micro check-in as illustrated in Figure 3.1.

There are several factors that should be considered before choosing one technology over others, as listed below.
Figure 3.1: Currently available technologies that can enable micro check-in.

- Cost: The adopted technology should not incur any costs to the users and it should be very inexpensive to use for the service provider.

- Convenience: The effort that the user needs to make to micro check-in must be minimal. At the same time, the setup for the service provider should be easy as well.

- Accuracy: The accuracy of obtained location must be high, since it is a primary requirement for micro check-in.

- User acceptance: Consumers must feel comfortable and be aware of the technology being used.

Keeping the above requirements in mind, the three most promising technologies that can be used to enable micro check-in are (a) image recognition, (b) near-field communication, and (c) QR codes. In the following section we discuss these technologies in the context of micro check-in.
3.3.1 Enabling Technologies

In this section we discuss the technologies that can enable the user to check-in at a finely defined location. This can be done in more than one way, and here we discuss the three most promising means. We also discuss the pros and cons of these technologies in the context of the requirements set forth previously in Section 3.2.

Image Recognition

Image recognition includes methods for acquiring, processing, analyzing, and understanding images [11]. Via the smartphone camera, a user can capture an image of an object or setup at the point of service. This image can be recognized by advanced image recognition algorithms, and thus it can be known by the authenticating party where the user is.

Pros: The advantage in this method is that there is no additional setup needed at the point of service. Any setup can be linked to a specific information or data, using an application such as LinkaPic [21]. This is well-suited to micro check-in applications related to artwork, exhibits, painting, wine label, and magazine, where there already is a specific visual clue.

Cons: Due to the fact that there is no special setup needed, users may not recognize the existence of a service point. Often users need a clue about the existence of a service point itself. This could lead to reduced usage. Another minor disadvantage of using image recognition for micro check-in is low customer awareness – that it is relatively less accepted among the consumers.
Near-field Communication

Near field communication (NFC) [19] technology was conceived to bring mobile devices and physical objects together. It is a wireless technology for simplified transactions and data exchange between two devices in proximity to each other (within few centimeters). NFC is an extension based on radio frequency identification tags (RFID) which are data chips that store unique identification information that then can be read by an external data reader through the use of radio waves. NFC enabled smartphones provide a contactless transaction in which the user only has to hold the smartphone near the reader to access the data on the card.

Pros: The technology provides high accuracy as well as convenience at the same time. If the user has an NFC-enabled device he or she can micro check-in very easily. Bulletin boards and smart posters are potential use cases.

Cons: There is a specific setup required on the part of the service provider, whose cost is not trivial. The main disadvantage is the lack of widespread use – at the time of writing, only a few commercially available smartphones have NFC microchips.

Quick-response Codes

A quick response code (QR code) is a two dimensional barcode designed specifically to be read by camera phones and dedicated barcode readers. Based on the type of data embedded in the code, it might direct the viewer to a website, make a phone call, or do other actions. The code consists of black modules arranged in a square pattern on a white background. Users have to download a QR code reader on their smartphone, but there are several free QR code reader
applications available for all smartphone platforms.

**Pros:** A QR code is based on an open format whose specification is available royalty-free. It is compact, error tolerant and responds quickly. The main advantage is that it has a growing acceptance among consumers. Although initial setup is required by the service provider, the costs are quite low (printing). Since most mobile phones have a camera, the technology is available to a majority of mobile phone users. The presence of a QR code also gives the user an indication of the presence of a service point.

**Cons:** One major disadvantage of QR codes is that it does not look beautiful. At many service points, the aesthetics of the arrangement is crucial and the service provider may not want to put an ugly-looking QR code there.

However, we believe that the advantages (namely, simplicity and wide acceptance) of using QR codes for micro check-in far outweigh the disadvantages, and hence, we chose QR code for OnTable, our micro check-in application for restaurant ordering. Our choice for QR code is just for the validation purpose and can be replaced by other technologies as needed.

### 3.3.2 The Setup

In the following section, we discuss the setup of a micro check-in system.

The general setup involved in a micro check-in system is illustrated in Figure 3.2. In a typical micro check-in interaction, there are several entities and concepts involved that have specific meaning in our context.

**Service provider:** The service provider, typically a business or similar entity, defines the service points where users could check-in. The service provider’s staff
Figure 3.2: Setup of a system that allows users to reveal location at specific service points to receive services from a service provider.

will see the service requests from users on their computers, following which they will prepare the order at a service fulfillment area and provision the service to the user.

**User:** The user is the customer who gets the benefit of the service. This benefit is either due to improved efficiency or convenience, or is due to an enhanced experience. The user will typically use a mobile device owned by themselves or by the service provider.

**Points of service:** These are specific finely-defined objects or places where
Figure 3.3: Functional block diagram of a general micro check-in based system.

the users check-in to obtain specific services from the service provider. Examples include an exhibit, a wine bottle, a restaurant table, and a gym machine. These points are within the *user-service interaction area*. The service point optionally has a visual *point of service indicator* or facilitator that lets the user know about the existence of a point of service. The service point typically is also the physical space where the service is rendered.

**Service requests**: The list of services that are requested by the user. This is chosen by the user from a wider list of all possible available services. For example, this could be the list of items that are ordered by a customer.

**User-service interaction area**: The system involves a specific designated
area that is typically under the control or administration of a service provider. This area is the user-service interaction space and is usually the physical space where the service is rendered to the user.

**Means of check-in**: This is the method or technology that a user employs to check-in to the point of service. We discuss, in Section 3.3.1, several technologies that can be employed to enable micro check-in.

### 3.3.3 The Workflow

In Figure 3.3 we provide various functional modules that are part of the system. The user first uses a means of check-in at a service point to reveal his or her location. Via a web server, this request goes to the database server, that accesses the data and provides a list of available services to the user. The user places a service request that goes to the service provider via a authentication module at the web server. The service provider processes the request at the service provider’s space and provisions the service. The service is typically processed at a service fulfillment area within the service provider’s administration space.

**Typical Software Architecture**: One or more web servers and one or more database servers drive the system for both the user and the service provider. A software application or a web application on a mobile device form the communication point for the user. A software application or a web application on the service provider’s computer receives the service request from the web server, and also assists in managing the progress in provisioning the service. Alternatively, the service provider can simply use a printer or a fax machine to obtain the service requests. The database server manages and maintains all the data pertaining to the application, including the list of rendered services, list of available services,
a log of users, and users’ favorite service items.

The workflow for the user is shown in Figure 3.4. The user uses a mobile device to reveal his or her location at the point of service via the optionally placed point of service facilitator. This action of the user allows him or her to see a list of available services. The user then chooses the services he or she would like to receive micro check-in using mobile device. The service requests are passed to a central web server. The web server authorizes and authenticates the user (Figure 3.5 step (e)) and forwards the requests to the service provider’s computer or device.

The workflow for the service provider is shown in Figure 3.6. The service provider’s staff sees the service requests on their computer as a notification. In
Figure 3.5: Flowchart describing the workflow of how a user reveals location at high resolution ("micro check-in") to receive service(s) from a provider at a point of service.

general, human staff may or may not be involved in rendering the service. The service requests are processed by the service provider at a service fulfillment area. Finally the service is provisioned to the user. We now discuss various technologies that can provide the means for micro check-in to the users.

### 3.4 Micro Check-in Applications

In this section we discuss several new applications that can be enabled by the micro check-in concept. One of the most promising applications of the micro
Figure 3.6: Flowchart that describes the workflow for the service provider.

check-in framework is for restaurant ordering by checking-in at a table. We will provide a detailed description, design considerations, and benefits of that application in Chapter 4. In Section 3.4.1, we briefly discuss a number of potential uses of the proposed framework.

3.4.1 Example Applications

In this section, we briefly discuss several other applications that follow the general setup described in Figure 3.2– Figure 3.3.

At a gym

Most modern treadmills and similar equipment allow users to view some essential statistics such as distance traveled, calories burnt and heart rate. However,
Manually keeping track of these statistics on a daily basis is difficult. Micro check-in can solve this problem by allowing users to check-in at a gym machine and getting daily statistics wirelessly. The gym plays the role of the service provider, their customers are the users, and the gym machines are the point of service. The gym staff would need to set up the system that can allow getting the statistics from the gym machine. QR codes or image recognition could be the appropriate enabling technology for micro check-in. Image recognition could be useful even in cases where the machine is not networked. The statistics could be captured from the picture of the information screen. QR code would be more appropriate when the machine is networked and the information about the machine ID and the time would be enough to get the statistics. This probably could also be used to manage requests for devices when there are more users than devices.

At a museum

With the emergence of digital technology, many museums have adapted the digitized way of sharing information via virtual exhibits. These digital exhibits have a plethora of information not readily available at the actual museum exhibit. With micro check-in, users can check-in at a museum artifact and get this information available digitally (history, video, brief documentary, wikipedia article, and so on). Users can also store the information, enjoy it later, and share it with friends and family. The museum staff is the service provider here, the visitors are the users, and the exhibits are the point of service. Many museums have already made efforts in creating enhanced content related to popular exhibits. As means of enabling micro check-in they simply have to place QR codes or in the case of image recognition, do nothing other than educating users. QR codes have the benefit of the users easily recognizing the location of the point of service.
In a classroom

Learning in a classroom can be made more interactive and fun if the students can obtain illustrations and other information in their mobile devices (e.g., smartphones or tablet computers) by micro check-in at a desk in the classroom. In this application, the students are the users, the desk is the service point, and the teacher (and/or the school) is the service provider. The classroom is the user-service interaction space.

At a mall or an amusement park

We often need to go to some specific shop or travel from where we are (i.e., we need specific directions within a mall or an amusement park). Currently there is no good way to get the directions other than looking at the maps inside the mall or the amusement park, because either the GPS does not work or the digital maps are not readily available on Google Maps or similar service. This problem can be addressed using micro check-in. The staff can define several points within the mall or amusement park where users can micro check-in (say, via QR codes) to reveal their location and in turn receive the directions to the place they want to be. In this case, the people are the users, the defined points are the service points, and the mall/amusement park is the service provider. The whole area within the mall could be the user-service interaction area. The means of revealing location is using QR codes, although image recognition or radio tags could be used in an alternative application.
In an airplane

When we are seated at a seat in an airplane, we often need to be served since getting up from the seat to request the service is not allowed or is not convenient. People have to wait to call an airplane host passing by, or use a button to request service. With micro check-in a user can reveal his location at a seat and request the desired service via the smartphone. The airplane host or hostess can see the request and provision the service in the most efficient manner. This saves time for both the air travelers and the service staff. In this application, the air traveler is the user, the staff is the service provider, the seat is the point of service, and the airplane is the user-service interaction area.

At a grocery or other store

We have to go through a lot of trouble when buying something at a grocery or general store. We have to punch loyalty cards as well as show a credit card in order to complete the purchase. This process can be simplified by using micro check-in at the check-out counter at the grocery or other store. The user can micro check-in (reveal his or her location) at the check-out counter to see the bill and pay using his or her mobile device (such as a smartphone). The loyalty card can be applied easily using the user's authentication information. In this case, the grocery store is the user-service interaction area, the customer is the user, the store is the service provider, and the means of micro check-in could be a QR code or NFC.
At a casino

People use loyalty cards at a casino and need to give the card for swiping everytime they sit at a casino gaming table. This inconvenience can be addressed by micro check-in at the gaming table. The casino staff can also provide drink service via this same service. Also, similar to the mall application mentioned above, the same application can provide directions to users when they are trying to find their way inside the casino. In this application, the casino visitors are the users, the gaming tables (or various points in case of directions application) are the service points, and the service being provided include loyalty tracking and/or directions or other experience-enhancing services.

At a sporting event

When we go to watch a game and need a drink or snack, we often have to interrupt the game to go and get it. By revealing their location (i.e., micro check-in) at the seat people can place orders for service such as drinks or snacks without going away from their seats. The staff will know exactly which seat number the order came from, and can fulfill the order right away and either serve the food/drink, or notify on the user’s phone when the order is ready for pickup. In this case, the audience is the user, the seat is the service point, the stadium is the user-service interaction area, and the catering staff is the service provider.

At a theater

The situation at a theater (movie or otherwise) is similar to a sporting event, where the audience can micro check-in to place and receive orders rather than going to the service station and standing in a queue to buy drinks or snacks. In
this application, the customers are the users, the seat is the point of service, and
the catering staff is the service provider.

The above list provides some of the promising applications of the proposed
framework.
Chapter 4

OnTable: Design and Engineering

OnTable is one of the most promising of many possible applications of micro check-in. OnTable helps customers order food using their smart phone while sitting at a restaurant table. As mentioned earlier, micro check-in is enabled by the user scanning a QR code available at the restaurant table. As discussed before, using a simple QR code scanner to enable micro check-in will provide low maintenance and efficient user experience. In this chapter we present the design decisions, business analysis, software architecture, and implementation details of OnTable.

We start by providing details of the OnTable application framework (Section 4.1). Next, in Section 4.2, we motivate the business case for this application. We then delve into the software design aspects and provide a detailed discussion on the architecture, setup and workflow in Section 4.3. In Section 4.4 we lay out the screen shots of the conceived application.
4.1 What is OnTable?

OnTable allows patrons to order food using their mobile devices while sitting at a restaurant table. This allows the user to place an order without directly interacting with restaurant wait staff, thus saving the time of both the restaurant and the user as a result of the improved efficiency. OnTable includes a mobile application (for the user) and a web application (for restaurant staff, the service provider). It uses a QR code placed at a table (the point of service) to retrieve the menu from a web server for a particular restaurant. This framework enables real-time ordering (providing enhanced, more efficient, service). The restaurant-admin web application provides an interface to support restaurant staff allowing them to process the order and to perform updates in the menu.

4.2 Business Analysis

OnTable provides several benefits to the restaurants as well as to the users. From a business perspective, the restaurants are the customers of the OnTable service, since they will be paying a nominal service fee for providing the enhanced ordering service to the users.

Benefits to the restaurants: OnTable provides great value by increasing customer satisfaction. Below is a list of potential benefits of using OnTable to the restaurants.

1. Quick turnaround: Due to improved overall efficiency, the restaurants can serve more customers in a given amount of time.
2. *Offer better service:* The restaurant staff is freed up from essentially repetitive work of taking order and punching into a POS system. This allows them to engage better with the customers.

3. *Fewer errors:* Since the order from the customer goes directly to the POS or responsible staff there are fewer avenues of error.

4. *Analytics:* Restaurants can easily know and analyze the dishes that customers like (add to their favorites) and share most often.

5. *Engage customers:* Restaurants have a platform to better engage with customers by pushing special offers and deals.

**Benefits to the users:** The end users of OnTable also get an enhanced overall experience, due to the following benefits.

- *Order when ready:* Customers will have an optimal ordering experience because they can order right when they are ready, not when the server comes to ask. Sometimes users are not yet ready, and sometimes they have been waiting for a long time.

- *Save or order from favorites:* Users often order a selected few dishes that they like. There is no easy method available today for them to keep track of their favorites. Now, with OnTable, users can not only keep track of their favorites, but are able to quickly order.

- *Share favorites and get recommendations:* Eating out can become even more social with the ability for users to share favorites with their friends, and in turn, get good recommendations from them.
- **Per dish ratings**: The rating sites popular today (e.g., Yelp) have a very crude rating system broadly averaging determining the quality of a restaurant. When ratings are available per dish (menu item) users can search for the item they crave and pick the best restaurant for that dish. This is also beneficial for the restaurants since there are some specific dishes that a restaurant specializes in, which can be better revealed using this system of rating rather than average rating for the restaurant.

- **Get specials and discounts**: Users can get specials and discounts, tailored to their favorites, right on their smartphone.

From the above discussion it is clear that OnTable can provide great benefits to both the users and the restaurants. We believe that there is a good business case for this application, and hence we move forward with designing an appropriate architecture and wireframes, which are discussed in the following section.

### 4.3 The Architecture, Setup and Workflow

OnTable has two user-facing components in its design. First, a smartphone application (Andriod and iPhone) that allow users to micro check-in via QR code and order food, and second, a web admin application for the restaurant staff that allows them to manage menu and process orders. There is a central web server that links the two. The mobile application communicates with the web server in real-time and exchanges the information related to orders placed from a table. The web server also communicates with the restaurant admin web application about the orders placed, and any menu updates.

Figure 4.1 illustrates the high level architecture of the system. The customer
scans a QR code to place an order using their smartphone. When the order is placed, the application calls the web server with the unique table id, time, date, and order details. After that, the web server validates the order by verifying the order information. To verify the information, the web server accesses data from the database server. If the order information is invalid, the web server rejects the order and sends a specific error message on to the user’s mobile phone. On the other hand, if the order information is valid, the web server calls the restaurant web application and provides valid data. The web application will then display the order information on the screen and indicate a new order request.

The OnTable system follows the setup that is described in Figures 3.2, 3.5, and 3.3. To illustrate how the general setup of Figures 3.2 and 3.3 is applied in
Figure 4.2: The setup for OnTable, a micro check-in application for restaurant ordering. Notice how simple adaptation of various blocks of Figure 3.2 allows description of this new micro check-in based application.

In the context of OnTable, we re-label the diagrams for this application and present them again in Figures 4.2 and 4.3. As can be seen from the figures, the general micro check-in setup can be easily adapted for this application. Such a re-labeling would form the basis of the design of other new applications as well.

Figure 4.3 illustrated the functional block diagram of OnTable. There are three major parts of the application including user space, web server and restaurant administration application. Within the user space, the user can perform two tasks. One is, a user can request for restaurant menu by scanning a QR code. Then that user space calls the web server with the menu request along
Figure 4.3: Functional block diagram of OnTable. Notice how the simple adaptation of various blocks of Figure 3.3 allows description of this new micro check-in based application.

with table id and restaurant name. The web server verifies the data and pulls the correct menu based on unique restaurant id/name and replies back. If the menu is already in the user’s smartphone’s cache, and there are no updates to the menu, it is not updated.

Another use case is, a user can select items from menu and place the order by scanning a QR code. This process sends the order directly to the web server. Then the web server authenticates the order information by pulling data from database server. After a successful authentication, the web server sends the order right to the restaurant administration application. The restaurant administration now sends the order to the restaurant kitchen. Once ready, the food is served at
the table.

4.3.1 Mobile Application

Figure 4.4 illustrates the mobile application flow diagram. Once the application starts, the users can browse the menu for nearby restaurants. The restaurant list is fetched based on the user’s current GPS location. While browsing the menu, users can put items on a tray if they like to order. After that, users can review the tray and finalize the order. To place the order, users micro check-in to a restaurant table by scanning a QR code available at a restaurant table. If there is an error during QR code scanning, the error message screen will be displayed and it will redirect the user to browse menu screen. Otherwise, users can submit the order by pressing a submit button on the screen. Users will get a chance to decide before finally placing an order. If they decide to order, on success the greetings message will appear on the screen and the application will end; otherwise users can go back to review the tray or select new items.

4.3.2 Web Application

Figure 4.5 illustrates the restaurant admin web application flow diagram. After a successful administration login, the home page for order information is loaded. If a new order arrives, the order color is changed and restaurant staff can verify the order information. After a successful verification of an order, the order information goes to the restaurant kitchen. Otherwise, the manual verification is done and then order information goes to restaurant kitchen. After a successful order, the wait staff serves the food at restaurant table and the order information is stored in a database.
4.4 Wire Frames for OnTable Application

Figure 4.6 illustrates the wire frames for the OnTable mobile application user interface. The mobile application interface is fresh and simple in functionality. The home screen displays restaurant names based on a user’s current GPS location. Users can select any restaurant and browse the menu. They can also select items and add them to their order tray.

Figure 4.7 shows an example of how to micro check-in to a restaurant table. To authenticate the user’s current location, a web server verifies the current GPS
Figure 4.5: Web application flow control diagram.

The user should enable his/her smartphone’s GPS to pass a web server authentication successfully.

Figure 4.8 shows how a user can store his/her favorite items using OnTable. The user can search the menu or browse the menu to add a favorite item. It can also be added from a tray. When a user is known to be at a given restaurant the location is verified by QR code check-in. Whenever the user is not at a specific restaurant, all favorites will be shown.

Figure 4.9 shows an example of placing an order from the tray using OnTable. While placing the order, a user is assumed to be already checked-in to a restaurant table, otherwise is prompted to scan the QR code. The user can add his/her favorites at any time. Once having decided to order, the user will see the order
Figure 4.6: A user can browse the selected restaurant’s menu on a smartphone. User can also look at the different categories of food in the menu. User also gets a chance to add items to the tray.

confirmation screen. The application will ask the user to confirm the number of people they are ordering for and the number of people at the table. This provides additional security to the application.
Figure 4.7: Micro check-in to a table: Once a user decides to place an order, he/she must scan the QR code placed on the table. The smartphone’s GPS should be enabled for additional security. A web server will verify the order request based on the user’s QR code data and the location data.
Figure 4.8: Favorites: The user can add a dish to his/her favorites. He/she can add notes and take a picture of the favorite-ed dish. He/she can also share it with friends on Facebook and/or Twitter.
Once micro checked-in to a restaurant, a user can place an order from his/her tray. He/she also has the option to add the selected items to favorites.
Chapter 5

Implementation and Results

In this chapter, we discuss the implementation of the OnTable application framework, along with the testing that we performed and the user validation that was carried out. We begin, in Section 5.1 with the implementation details, where we discuss the technologies used, and the specific user interface that was developed. Both the smartphone application and the restaurant web administration application are discussed here. Next, in Section 5.2, we provide details of how the applications were tested. We also present performance evaluation details for the overall system. We also conduct a comprehensive user validation for OnTable by carrying-out a fairly large survey. The details and conclusions of this user validation are presented in Section 5.3.

5.1 Implementation Details

This section outlines the specifics of how the OnTable framework was implemented. As shown in Figure 4.1, the OnTable implementation includes a web application and a mobile application. The web application implementation in-
cludes, login, add and update menu, creating accounts for restaurant staff, view transaction information and process orders. The smartphone application implementation includes, home screen, selected menu item tray, help screen and general navigation within the application. In the following sections, we present detailed description of each components (Section 5.1.1 and Section 5.1.2).

5.1.1 Web Application for Restaurants

A web application is accessed over a network such as the Internet or an intranet. Web applications are popular due to the web browsers and their availability everywhere, and the convenience of using a web browser as a client. There are two components of a web application (web service), namely the front-end and the back-end. Front-end is what the user can see and what the user interacts with. Back-end is the software that the user does not see. This is what works with the data behind the scenes. For example fetching, inserting, deleting, updating a database constitutes backend software code. We will describe both components in the context of OnTable in detail.

Integrated development environment (IDE): We used Aptana Studio 3 [22], an extension of Eclipse, as the IDE for the development.

Backend framework: For the web application backend, several popular choices of programming languages include Pearl, Ruby and Python. For OnTable, we decided to use Python. We employ the Django framework [4], which is a popular backend web framework written in Python. Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding make it suitable for rapid application development, as well as for use as
a scripting language. The core benefits of Python include, simple, easy to learn, easy to use, free and open source, portable, and extensible.

The Django framework follows the model-view-controller (MVC) pattern. MVC defines a way of developing software so that the code for defining and accessing data (the model) is separate from request routing logic (the controller), which in turn is separate from the user interface (the view). A key advantage of such an approach is that components are loosely coupled. In this framework and our implementation, the `models.py` file contains a descriptions of all database tables. The `views.py` file contains the business logic for the specific page. The `urls.py` file specifies which view is called for a given URL pattern (the routing). We have designed the back-end such that for each URL pattern, we have a view function to serve the request by either modifying the models or accessing data from models and return the response.

Our back-end uses JavaScript object notation (JSON) format to request/response data while communicating with the web and smartphone applications. Therefore, every time a request comes to a web server, we first deserialize the request object using Django’s Serializers library and use it. On the other hand, when preparing a response, we serialize the response object into JSON before wrapping it up into HTTPResponse response object.

**Backend deployment:** We deployed our web application on Amazon Web Services (AWS) that delivers a set of services that together form a reliable, scalable, and inexpensive computing platform “in the cloud”. We decided to use AWS because of its low cost, ease of deployment, and convenient environment. We used Amazon Elastic Compute Cloud (Amazon EC2) which is a web service that provides resizable computing capacity in the cloud. It is designed to make web-scale computing easier for developers. We use the nginx [16] web server with

**Front-end:** For front-end development, we use hyper text markup language (HTML), Cascading Style Sheets (CSS), and JavaScript. As stated earlier, the front-end implementation includes login, add and update menu, creating accounts for restaurant staff, view transaction information and process orders. In our implementation, the restaurant admin can login to the application and manage the daily changes to menu. The admin can create logins for restaurant staff so that they can login in to the application and view the orders and serve the customers.

Getting the response from the server can be done in two ways. One is to register a request to get new orders whenever the server gets a new order, and the other is to frequently ask the server to send new orders. Our web application makes frequent requests to the web server to get new orders or any update to pending orders. We set the interval of this request to a few seconds so that there is guarantee to get a quick response from the server. The **GET hyper text transfer protocol** (HTTP) request is made through asynchronous JavaScript and XML (AJAX) call to get all the orders from the web server.

The user interface of the front-end has two parts. The first part displays the number of tables via a rectangle that looks like a table. The second part is a small popup window for each rectangle to display order information. We distinguish the status of each order via a different color coding scheme for fonts. For example, a red colored font means new order, an orange colored font means the order is being prepared in kitchen, a green colored font means the order is served, a gray colored font means the bill is printed for the order and blue means the table is clean and ready to use.
When a GET request is successful, the web application displays the order information for each table within a particular restaurant. For example, if there are orders for table number 2 and 4 then the web application will display order information for those tables in red. The main page for order information is shown in Figure 5.1 and Figure 5.2.

5.1.2 Smartphone Application for Users

**Smartphone application:** For the smartphone application, there are several options available such as Google’s Android, Apple’s iPhone, Windows phone, BlackBerry, Symbian and HP’s Palm. Amongst these the most popular smartphones are Android and iPhone [1]. For OnTable, we are focussing on the two most popular smartphone platforms: Android and iPhone. For implementation of smartphone application, we used Eclipse as the IDE.

At the time of writing this thesis, we have finished the development of the
Figure 5.2: A popup window that shows the order information with item name, number of items and notes for a selected table.

Android application. Figure 5.3 shows the home page of the application which displays nearby restaurants based on the user’s current GPS location. The application shows menu items with price, a picture of that item, and a short description. Once the user selects a restaurant, they can browse through the menu to decide what they want to order. Figure 5.4 shows how a menu can be seen and browsed using the Android application. The add and delete buttons allows easy addition, and deletion of menu items to the tray (similar to shopping cart while shopping on Amazon). If a user wants to customize a specific item, they can click on the item and select customization options or write specific notes. The edit item page is shown in Figure 5.5.

When users are done with the selection of items to order, they can review the tray and submit the order by scanning a quick response (QR) code available at the restaurant table. After scanning a QR code, the confirmation message appears to make sure the user is ready to order because once an order is placed
Figure 5.3: The home page of the application showing example nearby restaurants based on user’s current GPS location

the user can not cancel it. The example tray is shown in Figure 5.6.

After placing an order, the user can go to a menu page and browse through it to order new items. They can select a few items and order again the same way we explained above. This time, only newly added items will be submitted.

5.2 Testing

In this section, we discuss the testing framework we used for OnTable architecture. We start with the testing framework for the web server in Section 5.2.1
followed by the smartphone application in Section 5.2.2.

5.2.1 Testing framework for Web Server (back-end)

As we mentioned in Section 5.1.1, OnTable’s web server (back-end) is implemented in Python on the Django framework. With Djangos test-execution framework and other utilities, one can simulate requests, insert test data, inspect applications output and in general, perform automated testing. We wrote unit tests for each of our Django views to test for correctness.

As per Django best practices, the testing code lies in tests.py file. We have
written functions to test for the connectivity, correct insertion and/or access to models (database tables) and to check for correct output. To simulate the requests, we first added some test data by writing a function to add random data to each models. We write assert statements to test for correctness of the output for each view function inside our `views.py` file.

5.2.2 Testing Framework for Smartphone Application

For the smartphone application, we used Java’s JUnit testing. The Android SDK provides tools and application programmers interfaces (APIs) necessary to
begin developing applications on the Android platform using the Java programming language. The Android testing framework provides powerful tools that help test every aspect of the application.

We used the JUnit TestCase class to do unit testing. The JUnit framework also offers Android-specific setup, teardown, and helper methods. We used the JUnit Assert class to display test results. The assert methods compare values we expected from a test to the actual results and throw an exception if the comparison fails. We did JUnit testing for our main activity class to test for connection, correct data input/output using assert statements.

Figure 5.6: A page showing a tray for a selected restaurant.
Besides JUnit testing, we also used *monkeyrunner*, which is a tool that provides an API for writing programs that control an Android device or emulator from outside of Android code. It’s a command-line tool for stress-testing UI by sending pseudo-random events to a device. The monkey tool runs in an Android Debug Bridge (ADB) shell directly on the device or emulator and generates pseudo-random streams of user and system events.

The monkeyrunner test included providing different numbers of generated pseudo-random events (such as 500, 1000, 1500) to check if the application breaks. The command we used is `adb shell monkey -p your.package.name -v 500`. Running this test on different versions of our code helped us to continuously debug the UI related problems and fix them easily by analyzing the log information.

### 5.2.3 Performance Testing for Web Server

We carried out performance testing for the OnTable application framework to measure the average delay starting from placing an order from the smartphone to getting the order at the restaurant admin web application. Our goal with the experiment is to monitor the server load versus latency/delay. Based on the experimental results we are able to measure web server performance and to set parameters such as the frequency of updates from the web application.

The experimental setup worked in the following way. The client simulation continuously sends orders to the web server. Once the order is sent, the web server validates the order request parameter and sends the order information to the restaurant admin application. On the other hand, the admin simulation periodically refreshes itself to get the new order request from the web server. This is done synchronously with a purpose of testing the server load and measuring
the delay between placing an order using client simulator and receiving the order at the restaurant admin simulator.

For this performance test, we conducted several time delay tests. In order to find the range of polling frequencies that provides a good trade-off between server load and time delay, we increased the order polling time interval exponentially and measured the average delay in seconds. Figure 5.7 shows the graph of average time delay in sec versus server load in requests per second. In the experiment, the admin simulator synchronously polls the web server for new orders at a regular interval. On the other hand, the client simulator continuously sends order requests to a web server. We observe not surprisingly that as the server load increases exponentially the delay also increases. We find that the interval
range of 100 milliseconds to 1 second is most suitable for the server load versus time delay trade-off that is practically workable.

Based on the above observation, we conducted another experiment specifically for the interval varied linearly starting with 100 milliseconds (msec) and up to 1000 msec, with the goal of finding the optimal interval parameter. Figure 5.8 shows the results of our experiments. The graph shows an average delay for each time interval. One point to note from the graph is that at some places the server load parameters gives constant delays (1.1 and 1.6 seconds). In the end, the delay increases significantly and reaches upto 1.9 seconds. From this we can conclude that the server performs better in certain cases. It is clear from the graph that we can find the range of order request time interval which should be acceptable.
to any restaurant.

From the experiments discussed above, we find the average time delay of 1 second to get the order request at the restaurant admin web application, which we believe would be acceptable to the restaurants. This average delay time is negligible compared to a regular order taking system. For future improvement, we believe that the server performance can be made better by refactoring the web server models written in Django. Our goal would be to improve the server performance and to make the communication asynchronous between the web server and a restaurant admin web application.

5.3 Results

In this section, we discuss the results of a usability survey that we carried out at two separate computer science upper level undergraduate classes at Cal Poly (Professor Franz Kurfess’s CSC 484 and SCS 486 classes). The purpose of the survey was to validate the hypotheses for OnTable that (i) it helps restaurant customers to order food utilizing their own smartphone without having to wait for the busy restaurant staff to come to them and take their order while sitting at a restaurant table, (ii) it helps reduce errors, and (iii) users like to use their smartphones at a restaurant to browse menu and place orders.

The usability study included 40 students. We selected a restaurant called Vieni Vai Trattoria, the first restaurant that we signed up for beta testing of OnTable.

For each class, we divided the total number of students into groups of 3. We gave a QR code to each students which contained a unique restaurant id and
Then we asked them to use OnTable to browse the menu, decide what they want to order, add items to the tray, review their tray and place their order by scanning a QR code. Once they place their order, we showed them that their order is directly displayed in the web application where the restaurant staff can see it and process it by sending the order to the kitchen. The web application looks similar to Figure 5.1.

We did the survey with the 40 students. We asked each student to fill out a survey to give us feedback with their experience of using OnTable. We asked them 14 diverse questions such as their likes and dislikes about features of OnTable, suggestions to add/update features in a current version, how much they like the concept of OnTable, their pain at different restaurants while ordering food, and
Amongst those 40 students, 33 filled out the survey. The results we got were quite positive and also very interesting. From the results, we found that the top two most frustrating problems encountered by the users in their eating out experience at a sit down restaurant are 1) waiting to order food and to get the food and 2) waiting for the check. The purpose of using OnTable at a restaurant is to eliminate the wait time to order food.

In Sections 5.3.1, 5.3.2, 5.3.3 and 5.3.4 we will discuss four key questions and the responses from the survey in detail.

5.3.1 Concept validation

The first question we asked in our survey is “In your eating-out experience at a sit down restaurant, what are some problems and/or frustrations that you typically encounter?”. The purpose of this questions is to validate the concept of OnTable. We stated a hypothesis that OnTable helps restaurant customers to order food without having to wait. One of the future goals is to integrate a payment system which allows customers to pay their bills using OnTable. Figure 5.10 shows the results we got from the 33 survey respondents.

It is clear from the graph that 85% of users are frustrated with “waiting for the check” and 76% of users are frustrated with “Waiting to order the food and to get the food”. The result helps us to validate the need for OnTable application framework.
Figure 5.10: Results for the question: “In your eating-out experience at a sit down restaurant, what are some problems and/or frustrations that you typically encounter?” The Y-axis represents number of people. People may select more than one checkbox, so percentages may add up to more than 100%.

5.3.2 User acceptance validation

The second question we asked in our survey is “How likely would you use this application if it is offered by a restaurant for ordering food from the table using your smartphone?”. The purpose behind asking this question is to know if customers will use such an application which allows placing an order at a sit-down restaurant using a smartphone. Figure 5.11 shows the results we got from 33 students.

The graphs show that, three-fourths of the respondents are either extremely
likely or very likely to use OnTable if available at a restaurant (48% of the users are likely and 27% of the users are extremely likely). This is very encouraging, and it provides a strong validation for OnTable. This data will also greatly help us convince restaurants to sign up for our beta testing version which will soon be available in four San Luis Obispo restaurants.

5.3.3 Efficiency validation

The third question we asked in our survey is “What would be more efficient to order food while sitting at a restaurant table? Order food using paper menu or
Figure 5.12: Results for the question: “What would be more efficient to order food while sitting at a restaurant table?” The Y-axis represents number of people.

Order food using smartphone (using OnTable)?”. The main purpose of asking this question is to know if the users feel comfortable using a smartphone to browse the menu. The purpose of using a smartphone is to increase the efficiency of accessing the menu, and placing an order without the help of busy wait staff at a restaurant. Figure 5.12 shows the results we get from 25 students.

The graphs shows that, 80% of users agree that it is convenient to use a smartphone while browsing a menu. Only 20% of users still want to use the paper menu (used in traditional restaurant ordering system). The results also help us validate the basic elements of the user interface we designed for OnTable. With this response (80% users preferring OnTable), we conclude that the users
are comfortable using the OnTable menu-browsing interface as compared to a paper menu.

### 5.3.4 Fewer errors validation

Another question we asked in our survey is “OnTable helps to eliminate possible errors by placing an order accurately using smartphone. How much do you agree with the statement?” The question helped us to validate our hypothesis that possible miscommunication and errors caused by such miscommunication can be significantly lowered using OnTable. Figure 5.13 shows the results we got.
from 25 survey respondents.

The graph shows that, 36% of users think that fewer errors are possible while using OnTable to place an order rather than using traditional ordering system. It gives us positive feedback that using a digital menu instead of a paper menu can help reduce the miscommunication errors. We note that this question just gives the user’s perception about error rates, and not actual comparison with actual error percentage happening at restaurants. This will be investigated in the future during our beta testing experiments.

5.4 Summary

In this chapter, we presented the implementation details and user validation results for OnTable, an application of the micro check-in framework proposed in this thesis. The OnTable framework, comprised of a smartphone application for the users, and a web application for the restaurants, has been fully implemented over the course of the thesis research. We conducted comprehensive testing for all the components and modules of OnTable at various levels (including unit testing, monkey-testing, and performance testing).

We also conducted user surveys to validate the various hypothesis about OnTable. The potential users placed orders using Android phones and could see the orders coming in on the web browsers. The users then were asked to fill out a short survey. The results of the survey are very encouraging and validated our hypothesis related to the usability, efficiency, reduction of error, and accessibility. It is quite encouraging that 75% of the participants are likely or extremely likely to use OnTable at restaurants if available. Now that we have a solid validation of user acceptance, we discuss our goals for beta release experiment in the
next chapter.
Chapter 6

Validation

In this chapter, we discuss our goals for OnTable’s beta release and metrics to validate the results for OnTable. We start by discussing our beta release setup and goals in Section 6.1. Next, in Section 6.2, we present the data gathering framework. Finally, in Section 6.4, we analyze some threats to the validity of our beta release experiment.

6.1 Beta Release Goals

Our goal for the beta release is to collect customer feedback and analyze benefits of OnTable to patrons as well as to the restaurant owners. At the time of writing, we have already signed-up four restaurants in San Luis Obispo, CA, USA, for a beta testing. For the beta release, we will provide a soft copy of QR codes generated for restaurant tables. The QR codes have unique table ids, and restaurant ids embedded. The restaurants will put a hard copy of the QR code on each table. They will get access to their administration web account, where they can view and process the orders, create staff accounts, add or update menu
items, and pull up a daily transactions list.

6.2 Data Gathering Framework

The data gathering framework includes a number of experiments set up to collect customer feedback. They are explained below.

- Pick a busy day of a week in a restaurant using the traditional ordering system and gather data as per defined metrics.

- The next week on the same day, use OnTable and gather data as per defined metrics. Also, we can observe a few tables using OnTable and a few without using OnTable.

We have to repeat the above framework for several weeks to collect the required data. By repeating it every week, we will get accurate result to measure the difference between ordering food with/without using OnTable. We will also gather data about how many customers used OnTable on a specific day compared to using a traditional ordering system. This data will allow us to validate the concept and the hypothesis we made throughout this thesis.

The framework allows us to compare benefits of OnTable with a traditional restaurant ordering system. Another framework is:

- Monitor a restaurant with a traditional ordering system for a week, and gather data as per defined metrics.

- The next week, monitor a restaurant using OnTable and gather data as per defined metrics.
6.3 Metrics

Table 6.1 shows metrics used to measure the desired results of OnTable, which are discussed in more detail below.

Table 6.1: Validation metrics and how we measure it.

<table>
<thead>
<tr>
<th>Metric name</th>
<th>Explanation</th>
<th>Measurement approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turnaround Time</td>
<td>Number of customers a restaurant can serve</td>
<td>Difference between ordering food using OnTable and without OnTable</td>
</tr>
<tr>
<td>Accessibility</td>
<td>Quick availability of restaurant menu</td>
<td>Time difference between getting a paper menu and digital menu on smartphone</td>
</tr>
<tr>
<td>Order Accuracy</td>
<td>Smaller number of possible errors</td>
<td>Customer satisfaction with digital menu</td>
</tr>
<tr>
<td>Queue Time</td>
<td>Eliminates wait time to be seated at a table</td>
<td>Total number of table turnarounds</td>
</tr>
</tbody>
</table>

- **Turnaround Time**: The turnaround time can be measured by comparing the time difference between ordering food using OnTable and ordering food without using OnTable 6.2. The start action of ordering is when a customer sits down at a restaurant table. The end action of ordering is when a customer is done eating. If the food ordering with OnTable is faster than the traditional approach it proves that OnTable helps to improve turnaround time at a restaurant.

- **Accessibility**: The accessibility measurement includes the time difference between getting a paper menu and a digital menu on smartphone. However, the time to install an application on a smartphone can be negligible as it happens only once. With OnTable, users can look at the menu anytime even before actually going to a restaurant. The flexibility of browsing the menu anytime helps users to decide what to order even before sitting at a
table. Therefore, it eliminates wait time for getting a paper menu.

- **Order Accuracy**: The chances of errors are higher with a person taking an order (especially if there is a language problem). However, OnTable allows users to browse the menu on their smartphone which reduces possible communication errors and assures a more accurate ordering system. It also allows order flexibility as customers can customize their dishes by selecting predefined customization options for each item in the menu.

- **Queue Time**: OnTable eliminates queue time, that is, waiting time for users to be seated at a restaurant table during peak hours. It can be measured by comparing the total number of table turnarounds. On the other hand, if the number of customers are more than the restaurant can server, there is nothing can be done.

The above metrics will help analyze the benefits of OnTable to patrons as well as to restaurant owners.

### 6.4 Threats to Validity

There are some threats to the validity related to the beta release experiment. First of all, conducting the beta release in San Luis Obispo restaurants may or may not be representative of the results in general.

If a restaurant is using OnTable, wait staff will have more time to serve customers rather than taking orders and punching them into point of sales system. As a result, if at any time, customers need attention to choose dishes, the wait staff can help them quickly. This customer satisfaction may or may not be readily measured in our experiment. Also, the task of convincing restaurants to use the
proposed application in place of, or in addition to their currently used POS system can be challenging.
Chapter 7

Future Work and Conclusions

7.1 Future Work

Micro check-in has various potential markets as discussed in Chapter 3. Our future plan is to create a generalized platform for micro check-in that allows seamless development of other applications that are enabled by micro check-in. A generalized back-end will ease the development of other applications without duplicating efforts.

From a software development perspective, we plan to integrate an automatic credit card payment system with OnTable. This will provide flexibility for users to pay using their smartphone. Automatic payment completes the loop and makes it convenient to both the user and the restaurant. We also plan to integrate OnTable’s application framework with popular restaurant POS systems. Because of wide the adoption of POS in today’s modern restaurants, integration using the POS systems’ application programmers interface (API) will greatly reduce the operation burden for the restaurant owners and staff.
7.2 Conclusions

In this thesis, we proposed micro check-in, a new framework for interacting with the real world using smartphones. This opens up several applications that could utilize the power of smartphones in new ways, by letting people reveal their location at a finer resolution than possible by the popular GPS-based technology. OnTable is one of the most promising of many applications of micro check-in.

We believe that OnTable helps realize the benefits of micro check-in to order food from a restaurant table. It will provide quick turnaround time, increase the order accuracy, reduce the queue time, and improve overall customer experience. Using simple QR codes to enable micro check-in leads to a low maintenance and low cost solution for the restaurants. Also, the users can easily recognize the service points (the table, in this case) and do not need any specialized hardware to check-in. In summary, with OnTable, we are enhancing the conventional restaurant ordering workflow by providing a more efficient and user-friendly service.
Bibliography


[5] Facebook. Let people know where you have been, where you are headed and where you are now. http://www.facebook.com/about/location, 2011.


[26] N. Vrcek, G. Buba, and N. Bosilj. User acceptance of location-