**ITCS​ ​6122**

**SOFTWARE​ ​SYSTEM​ ​DESIGN​ ​AND IMPLEMENTATION**

**PROJECT REPORT**

**ON**

**PROXIMITY MARKETING SYSTEM USING BEACONS**

**TEAM​ ​MEMBERS**

**Aravind**

**Chandu**

**Harsha**

**Sai Yesaswy Mylavarupu**

**Venkata Ratna Ushaswini Vinnakota (800983632)**

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TOPIC** | **Page No.** |
| 1 | Abstract |  |
| 2 | Introduction |  |
| 3 | Materials and Methods used |  |
| 4 | Software, Hardware and Technical Requirements |  |
| 5 | UML Diagrams |  |
| 6 | Coding |  |
| 7 | Sample Runs with output |  |
| 8 | Results and Analysis |  |
| 9 | Conclusion |  |
| 10 | Future Enhancements |  |
| 11 | References |  |

**ABSTRACT**

We have built a **new system** that can be used by storekeepers for digital and proximity marketing. Admin can manage store keepers and regions. Store keepers can then publish deals in his store within a region. User who has registered for the app, can see list of deals in his proximity as defined by beacons’ region. User can also mark a region as his favorite and receive an instant push notification whenever a storekeeper in that region publishes a new deal.

System requires developing a web service for API (Application Programming Interface) endpoints that can be accessed by both web app and mobile app. Web application is required and accessed by admin and storekeeper to manage marketing. Mobile application is required to communicate with beacon and update user accordingly. Communicating with beacon brings context awareness to system.

Admin can login (with pre-specified credentials by system) and manages regions and store keepers. Whenever a new physical beacon is installed, admin can add that region to system. Whenever a new storekeeper wants to be included to a region, admin can add him to system and provide him login credentials. Storekeeper with the given credentials, can login and manage deals in his store. He can add new deals and edit or delete old deals. User who marked this region as favorite receives push notification. User who has mobile app installed and Bluetooth enabled, can register and will be all set to experience proximity marketing.

**INTRODUCTION**

Gone are the days when retailers could only rely on print, TV, and radio advertisements to drive customers into their stores. Now, shoppers expect more tailored messaging, with personalization that makes sense. Not only can beacon technology provide that, it can trigger the ad based on a customer’s proximity to the store. This helps troubleshoot some of the physical limitations of brick-and-mortar stores. Now, shoppers can be marketed to at the right time and in the right place.

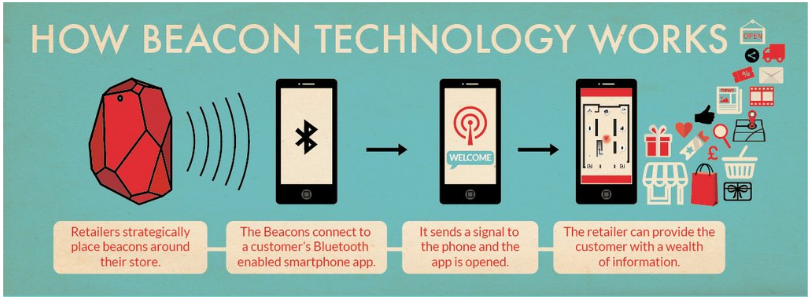
It is a very common situation that might happen to us when we visit a region/mall buying in one store and be completely unaware of deals in store right next to it. Proximity marketing deals exactly with this issue and uses cellular technology to send marketing messages to mobile-device users who are in close proximity to a business.

**Beacons for proximity marketing**

Beacons are small devices that transmit signals to devices, such as smartphones, when they are located within a certain range. Beacons are also ideal for driving customers into a venue or store to take advantage of offers or competitions that are only available instore.

They provide businesses with the ability to make transmit relevant, targeted information and messages to nearby devices. The beacons are used to send messages to apps when they enter the beacons range, which then update users. Once a physical beacon has been installed, all users that possess the necessary app and have their Bluetooth switched on can be marketed to via that beacon.

Beacon Marketing provides businesses with the unique opportunity to leverage their customers physical context by delivering targeted messaging based on where they are and what they are doing at that time. Compared to some other tech hardware and software, beacon technology is affordable to adopt.



**MATERIAL AND METHODS USED**

We have used ASP.NET Web API, Identity, Entity and Firebase cloud messaging (FCM) frameworks to develop Web service, Xamarin and FCM to develop Android mobile app, HTML, CSS, JavaScript, Bootstrap, jQuery and Java Servlets to develop Web Application.

#write here about mvc and figure from nbad

**Materials Used:**

* ASP.NET Web API
* Identity framework
* Entity framework(ORM tool)
* Firebase cloud messaging
* Xamarin
* HTML
* CSS
* JavaScript
* Bootstrap
* jQuery
* Java Servlets

**Methods Used:**

* # has been used for building the functionality of the web application and integrate with the front end and the database.
* Used HTML for designing the content of web-pages.
* Used CSS for styling of the web pages.
* Used JavaScript for controlling the behavior of the web pages.
* Used Bootstrap, which includes the HTML, CSS based templates and for making the website responsive.
* Used jQuery, which is a JavaScript library
* Used SQL database to store all the information of authentication, users, admin, storekeepers, regions and deals
* Used cloud platform – Azure to host web service and SQL database; Amazon Web Service (AWS) to host Web application
* Used Xamarin and FCM to develop mobile application using C#
* Used Identity framework to manage role-based authentication
* Used Entity framework – Object Relational Mapping(ORM) tool to manage data connections
* Used ASP.NET Web API to develop web services to be used by web and mobile apps

**Software Methodology used: Agile**

Agile is a software development methodology used to build a software incrementally using short iterations of 1 to 4 weeks so that the development is aligned with the changing business needs. It is a process which values individuals and interactions over processes and tools.

Agile methodology has two important features namely User stories and iterations. User stories provide simpler ways of capturing user requirements in a project instead of having a lengthy documentation model. In the iteration period, development of user stories happens.

As the number of developers in our team is very small, we tried to get a firsthand experience with agile methodologies instead of using traditional software development models like waterfall and spiral models. We got a better opportunity in understanding the various aspects of Agile methodologies.

We created many user stories for development through Agile approach. We worked on each of these stories considering one at a time. Some of the user stories are as follows- Implementing roles-based authentication for Admin, Store-keeper and User, Developing API end points to be accessed by Web and Mobile apps, Implementing Manage Store keepers functionality for Admin, Implementing Manage Locations for Admin, Implementing Manage Deals for Store keeper, Designing and Developing Mobile App and Implementing push notification.

Smaller teams always have a problem when the project is heavily dependent on documentation. Agile methodology helps to overcome this problem by providing emphasis on user stories.

**Project management tools:**

* Github to manage development and documentation
* Hangouts and In-person meetings for communication

**SOFTWARE, ​ ​HARDWARE​ ​AND​ ​TECHNICAL​ ​REQUIREMENTS**

**Software Requirements:**

* Visual Studio
* SQL Workbench
* Postman
* Eclipse
* # any web app specific

**Hardware Requirements:**

* Android phone to view mobile app
* Any user with a computer or smartphone with browser installed can view the web application
* Postman can be used to view/test API end points

**Technical Documentation:**

1. ***Web service development***
2. **Programming language**: C#
3. **Reused Algorithms and Programs**:
   1. We used Identity framework to manage roles based authentication and authorization.
   2. We used Entity framework (ORM tool) and followed Code first approach to design and develop tables in SQL database.
   3. We used ASP.NET Web API to develop API (Application Programming Interface) end points for the web or mobile application to access.
4. **Tools and Environment:** The complete web service project is implemented using Visual Studio-2017. We used Nuget package manager to manage nuget package installations of Entity and Identity frameworks.
5. **Database system:** We have used SQL database hosted in Azure. The name of the database is DiscountNotifier1\_db. Web app service is also hosted in Azure.
6. **Files**
   1. Functionality for the web service has been achieved through API controllers:
      1. AccountController.cs
      2. DiscountsController.cs
      3. RegionsControoler.cs
      4. UsersController.cs
   2. Models for code first approach to design tables are
      1. Discount.cs
      2. Region.cs
      3. IdentityModels.cs
      4. AccountBindingModels.cs
   3. Config file for web service – Web.config

#Write here about web app development

1. ***Mobile app development***
2. **Programming language**: C#
3. **Reused Algorithms and Programs**:
   1. We used Estimote SDK to communicate with beacons
   2. We used Firebase Cloud Messaging API to receive push notification when a new discount is published in user’s favorite region
   3. We used Xamarin to develop native mobile app using C#
4. **Tools and Environment:** The complete mobile app project is implemented using Visual Studio-2017. We used Nuget package manager to manage nuget package installation of Estimote framework and other required packages.
5. **Files:**
   1. Functionality of receiving push notifications is implemented using
      1. MyFirebaseIIDService.cs
      2. MyFirebaseMessagingService.cs
   2. Models to show data to user are
      1. Region.cs
      2. Discount.cs
      3. IdentityModels.cs
   3. Config file for push notification – google-services.json

**UML DIAGRAMS**

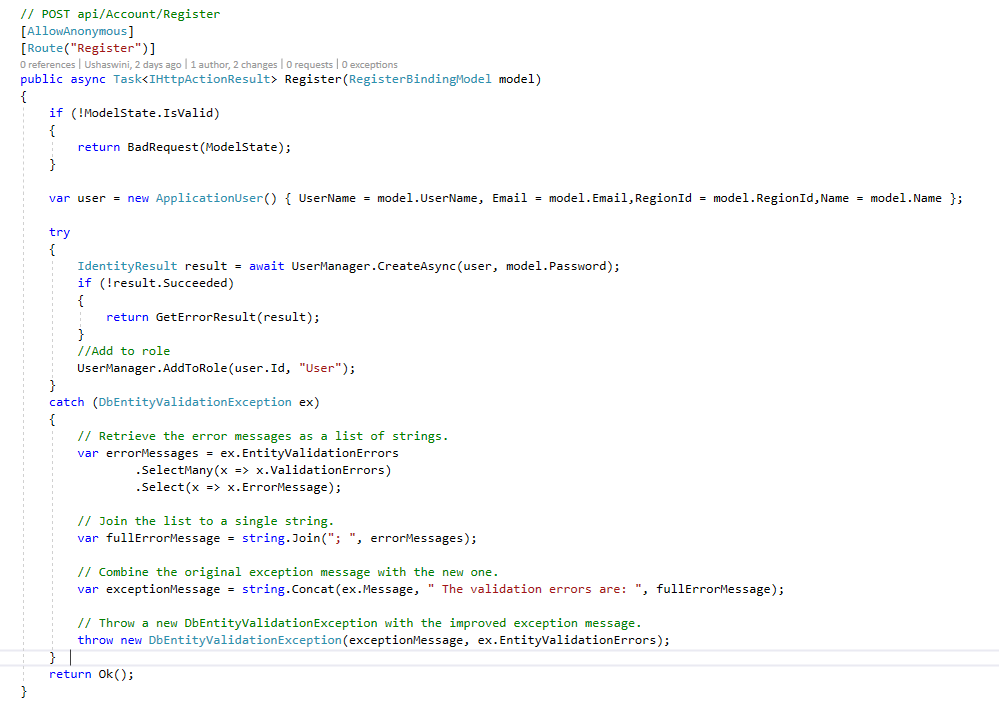
**#insert figures here**

**CODING**

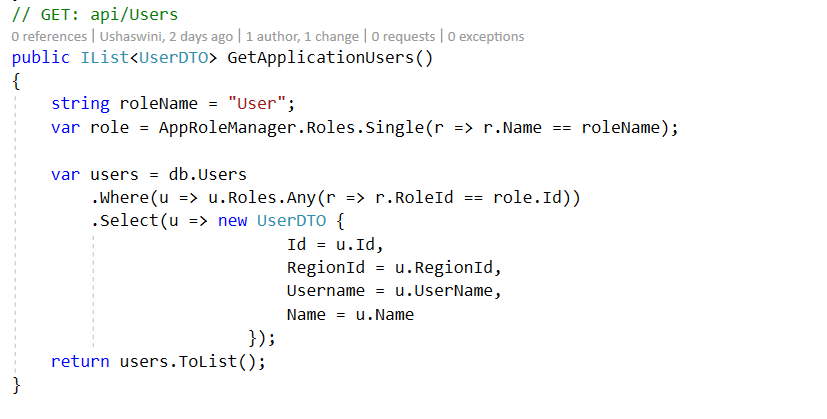
**API endpoint to add storekeeper by Admin**

****

**API endpoint to register as user**

****

**API endpoint to get list of Users – Data Transfer Object to avoid sending sensitive information like passwords**

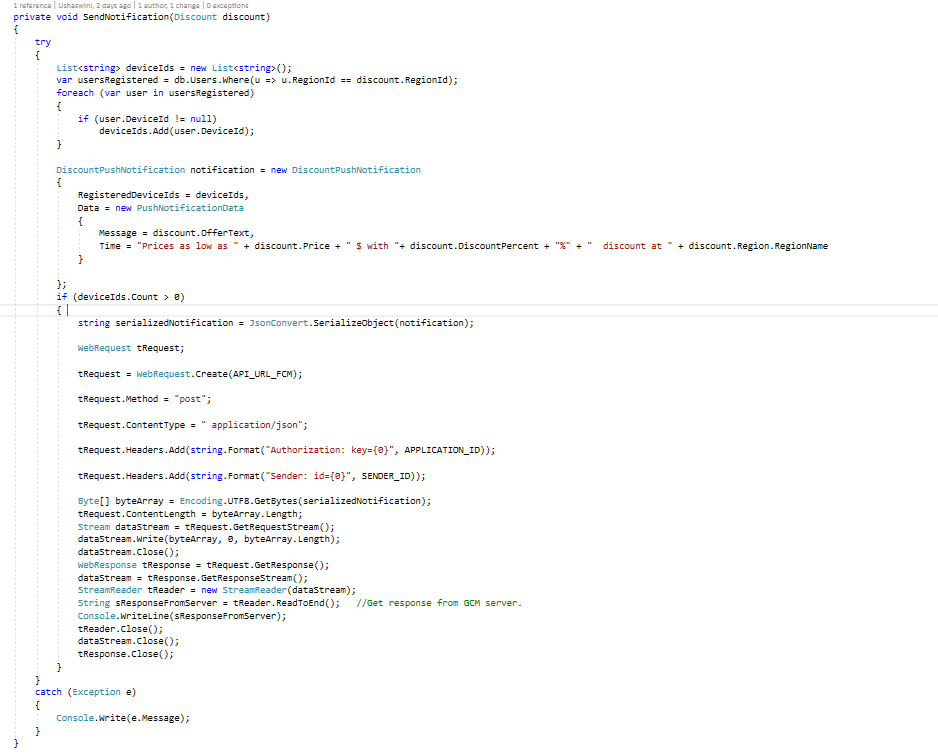
****

**API endpoint to update user’s favorite region**

**API endpoint to update device id to receive push notification**

**Service in mobile app to receive push notification**

****

**Method to send push notification every time a new deal is added to database**

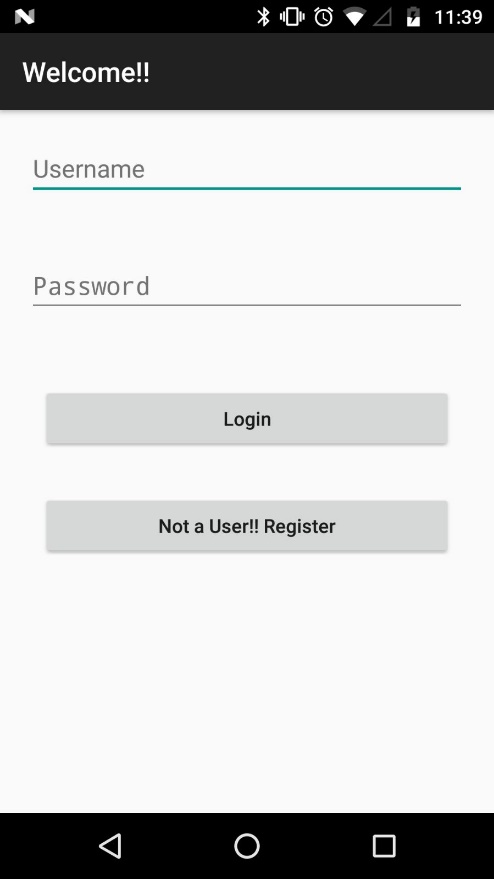
**#Add web app specific code**

**SAMPLE RESULTS AND OUTPUTS:**

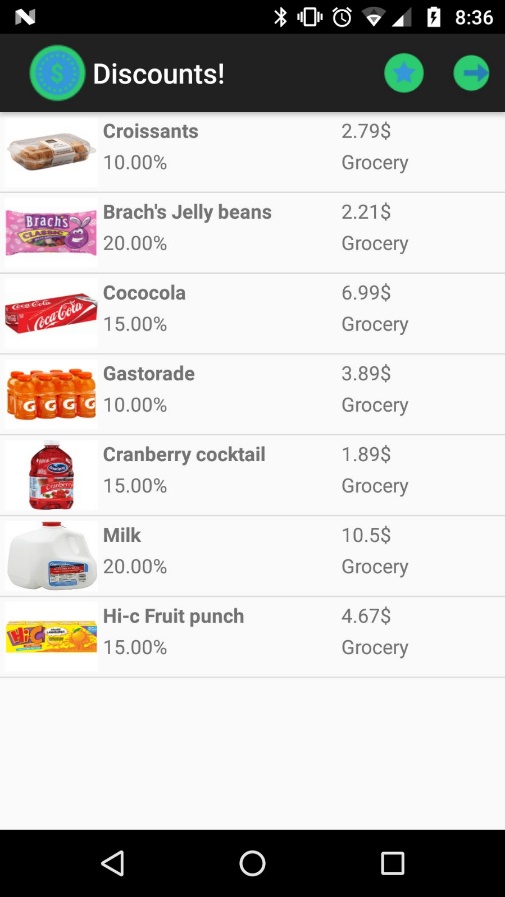
**#Web app pictures**

**Mobile App**

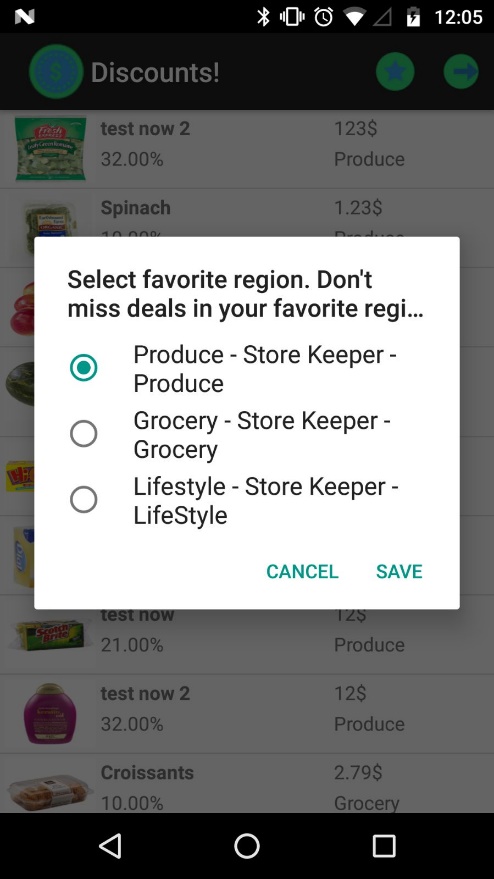
* **Login screen**

****

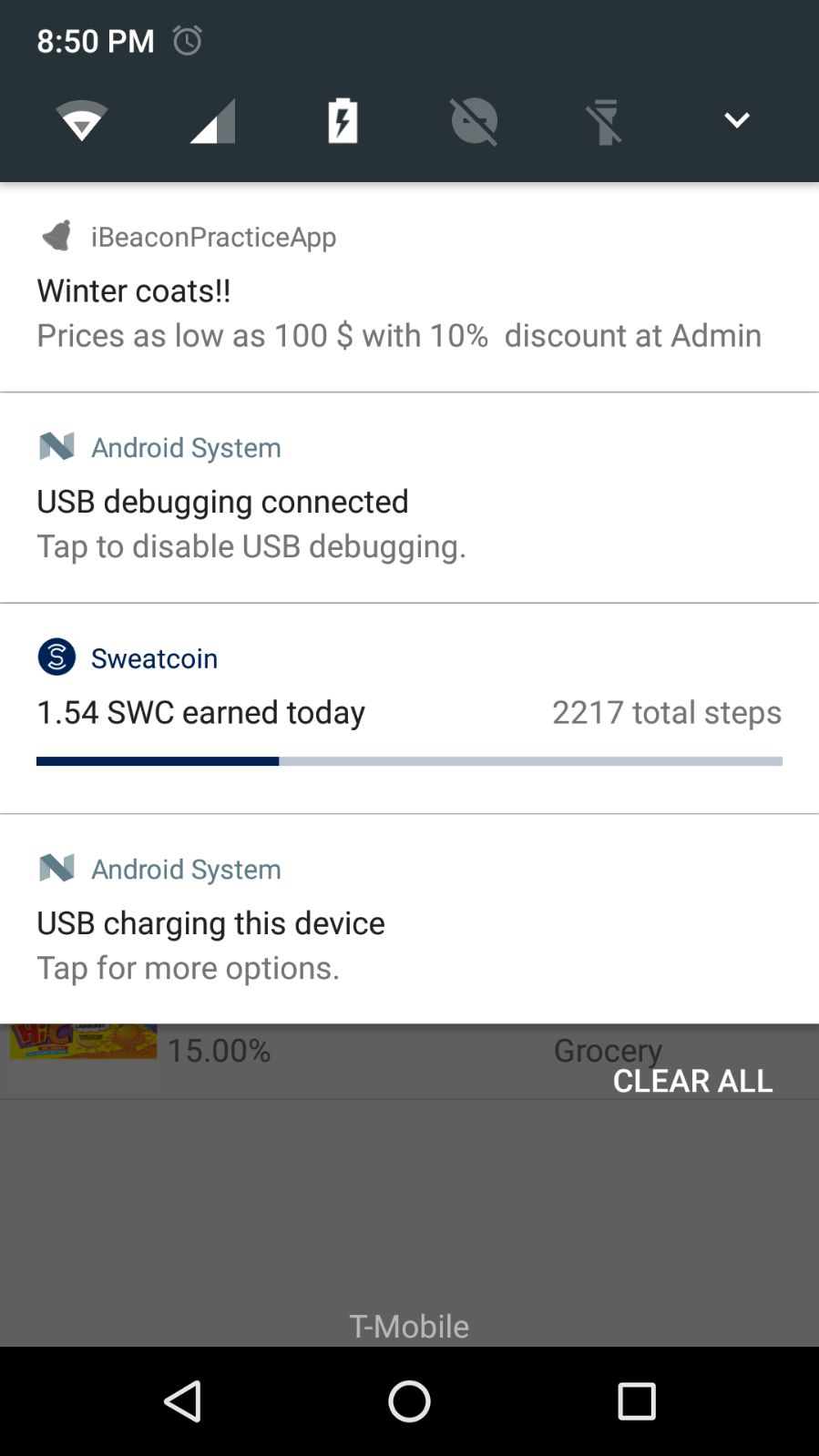
* **Dashboard screen**

****

* **Updating favorite region screen**

****

* **Receiving push notification when a deal is published in favorite region**



**CONCLUSION**

With more than six billion mobile phones in the hands of consumers today and with the numbers steadily climbing, phones are becoming a necessity of the 21st century. This makes just about every consumer with a smart phone potentially susceptible to a proximity marketing campaign, particularly younger, tech-savvy shoppers who are more likely to own smartphone and use them while shopping.  One of the key benefits of proximity marketing is that its messages reach a high percentage of potential customers who are in the area.

Beacon technology has proven to be effective as the bridge between brick-and-mortar retailers and customers glued to their phones. It’s a new way to engage with consumers and provide incentives for shoppers to enter a store, stay for a while, and hopefully make purchases. Compared to some other tech hardware and software, beacon technology is affordable to adopt.

This system targets proximity marketing using beacon technology by building web and mobile applications.

**FUTURE ENHANCEMENTS:**

* Allow user to opt-in and opt-out of notifications; At present user need to logout to opt-out from receiving notifications
* #any future enhancements

**REFERENCES:**

<https://firebase.google.com/docs/cloud-messaging/>

<https://www.shopify.com/retail/the-ultimate-guide-to-using-beacon-technology-for-retail-stores>

<http://www.marketing-schools.org/types-of-marketing/proximity-marketing.html>

<https://www.asp.net/web-api/overview/getting-started-with-aspnet-web-api>