Software Requirements Specification

for

PartyGuard Web Application

**Version 1.0**

**Prepared by Team 02**

**Northwest Missouri State University**

**07/19/2016**

**Table of Contents**

**Table of Contents**

**Revision History**

**1. Introduction**

1.1 Purpose

1.2 Intended Audience and Reading Suggestions

1.3 Project Scope

1.4 References

**2. Overall Description**

2.1 Product Perspective

2.2 Product Functions

2.3 User classes and Characteristics

2.4 Operating Environment

2.5 Assumptions and Dependencies

**3. Requirements**

3.1 Functional Requirements

3.2 Software Requirements

3.3 ER Diagram

3.4 Flow Diagram

3.5 Communications Interfaces

**4. Initial UI design screens**

# Introduction

**1.1 Purpose**

The purpose of this document is to provide software requirement specifications for the PartyGuard web application.

**1.2 Intended Audience and Reading Suggestions**

The intended audience are the host users/fraternity heads, master users who uses the PartyGuard web application. In contrast the basic user and the guard user who can perform basic functionalities through the web application can understand this document.

**1.3 Project Scope**

The project is a web application which will be accessible to three kinds of users (i,e Master user, Host User and Basic user). This application will offer service to subscribed fraternity houses only. Initially, fraternity houses will subscribe to PartyGuard for the service, then the subscribed party can use PartyGuard in their social gatherings to create a safe environment. Product includes PartyGuard web application, Android application and iOS application. However, this document will mainly focus on the web application scope.

## 1.4 References

The project proposal statement:

*<https://drive.google.com/open?id=0BzoVC8zt1fvUU3Znb3lTa1ptcFk>*

# Overall Description

## Product Perspective

The product being developed is a new, self-contained product. The context of this product is to provide protection to women who attends parties conducted by the fraternities in the university.

The overall system is when the user logs in through the mobile application and raises an issue, the guard user assigned by the host will react. The guard user comes to the location of the issue happened in order to rescue the women.

## Product Functions

The major functions the product must perform or must let the user perform can be well described in the form of a flow diagram.

## User Classes and Characteristics

There are four classes of users, they are

1. **Master User:** He can register a new customer (Fraternity) and he will give some credentials to Host User which are used to login after paying the Subscription fee.
2. **Basic User :** She is an actual user who uses Mobile Application to register ,Update the profile, change the password, Raise an issue and provide the feedback for resolution whether it is solved or not.
3. **Guard User:** This Guards will be allocated by fraternity head i.e. Host User who acknowledges the issues and solve them so that security will be provided to Women.
4. **Host User:** He is the head of Fraternity who can update the Fraternity house location if the house location is changed and finally he can also remove the members from the Fraternity*.*

## Operating Environment

It is compatible with all operating systems and all web browsers.

## Assumptions and Dependencies

In the client requirements, the host creating the events was not stated. So we have come up with the assumption that the host user can create a new event regarding a particular fraternity. Moreover the user can assign guards to that particular event. These details will be reflected in the mobile application. This feature will make sure that the basic users can view the event details and the guard users that are assigned to that particular event.

The dependencies are regarding the data that needs to be exchanged between the mobile application and the web application. The web application needs to get the basic/guard user data from the mobile application as the user registration happens only through mobile application. Furthermore the mobile application needs to get the data regarding the events and the guard users that was updated through web application.

# Requirements

## Functional Requirements

## Web Application will have a super user who will act as an admin and responsible for creating fraternity clubs.

## Fraternity clubs need to pay service fee to get party guard service.

## Party guard admin provide admin privileges to the fraternity club heads to be able to create events and to add in charges to that event.

## Fraternity club head can request for adhoc reports at any time.

## Software Requirements

**User Interface:** Material design for Bootstrap, AngularJS

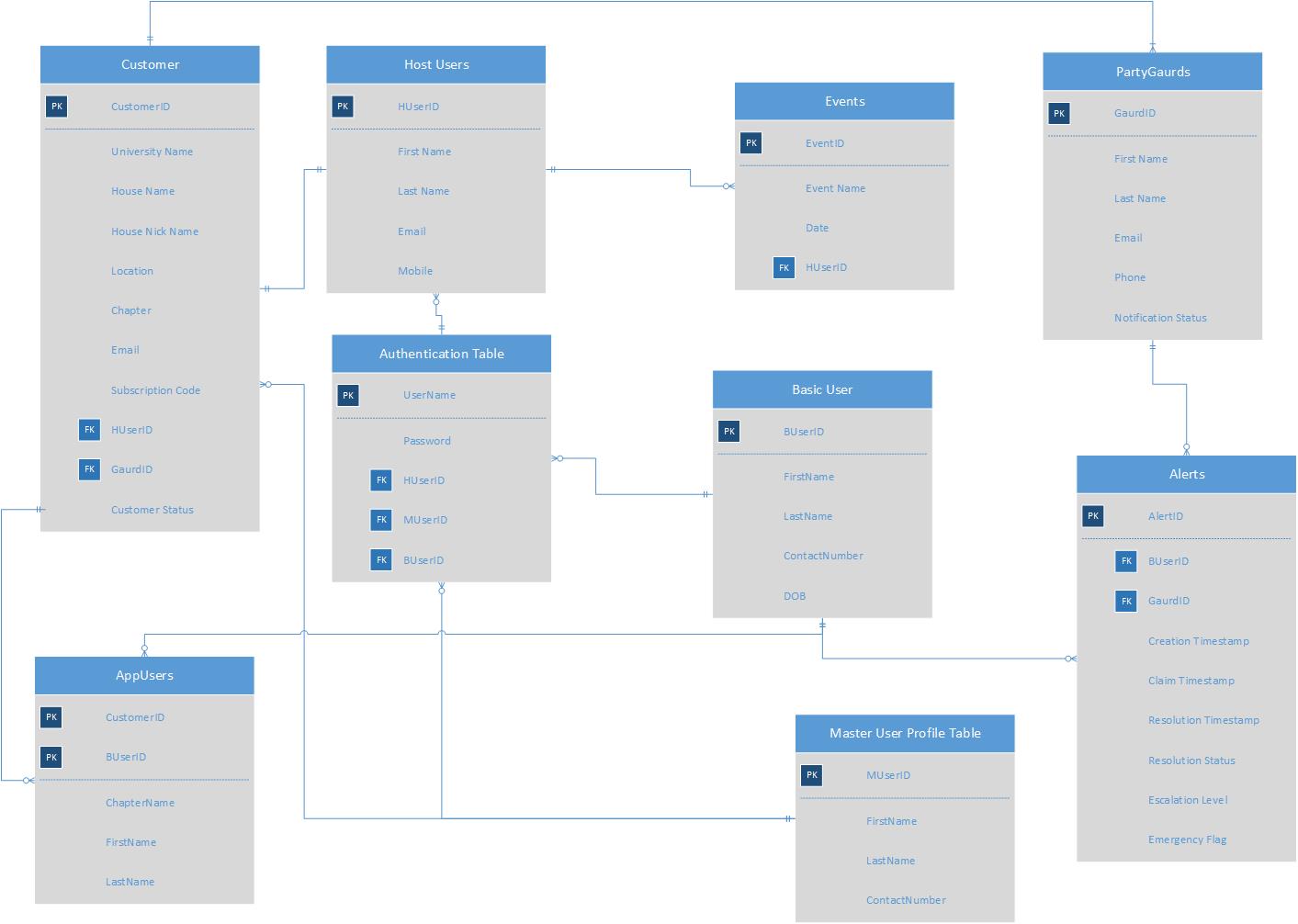
**Backend:** JAVA, RESTful API, Hibernate

**Database:** AWS

**Version Control:** Git using GitHub

**Issue Tracking:** GitHub

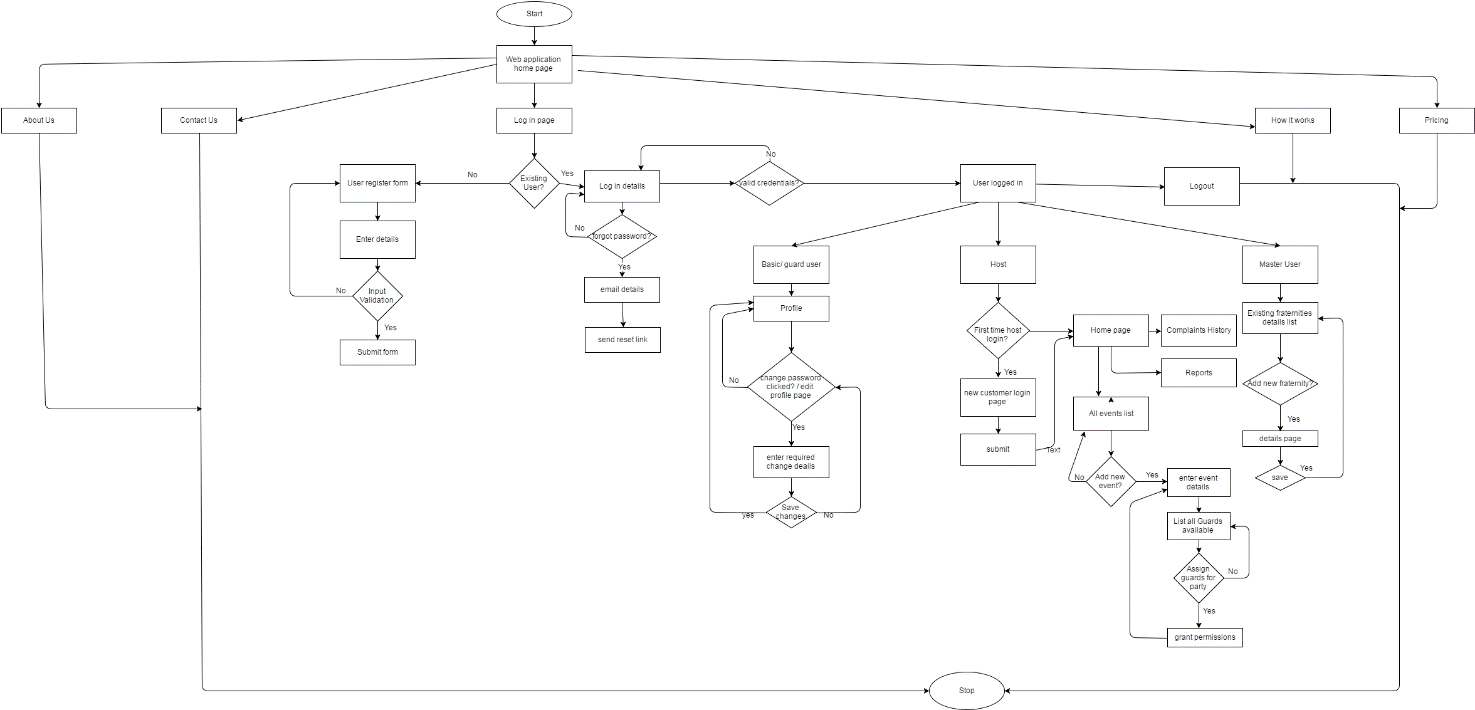
* 1. **ER Diagram**

****

**Figure 01: ER Diagram**

To have a clear view of the ER diagram a separate Visio file has been attached in the folder.

* 1. **Flow Diagram**

****

**Figure 02: Flow Diagram**

In order to have a clear view a separate xml file has been attached in the folder. Open the file using <www.draw.io>

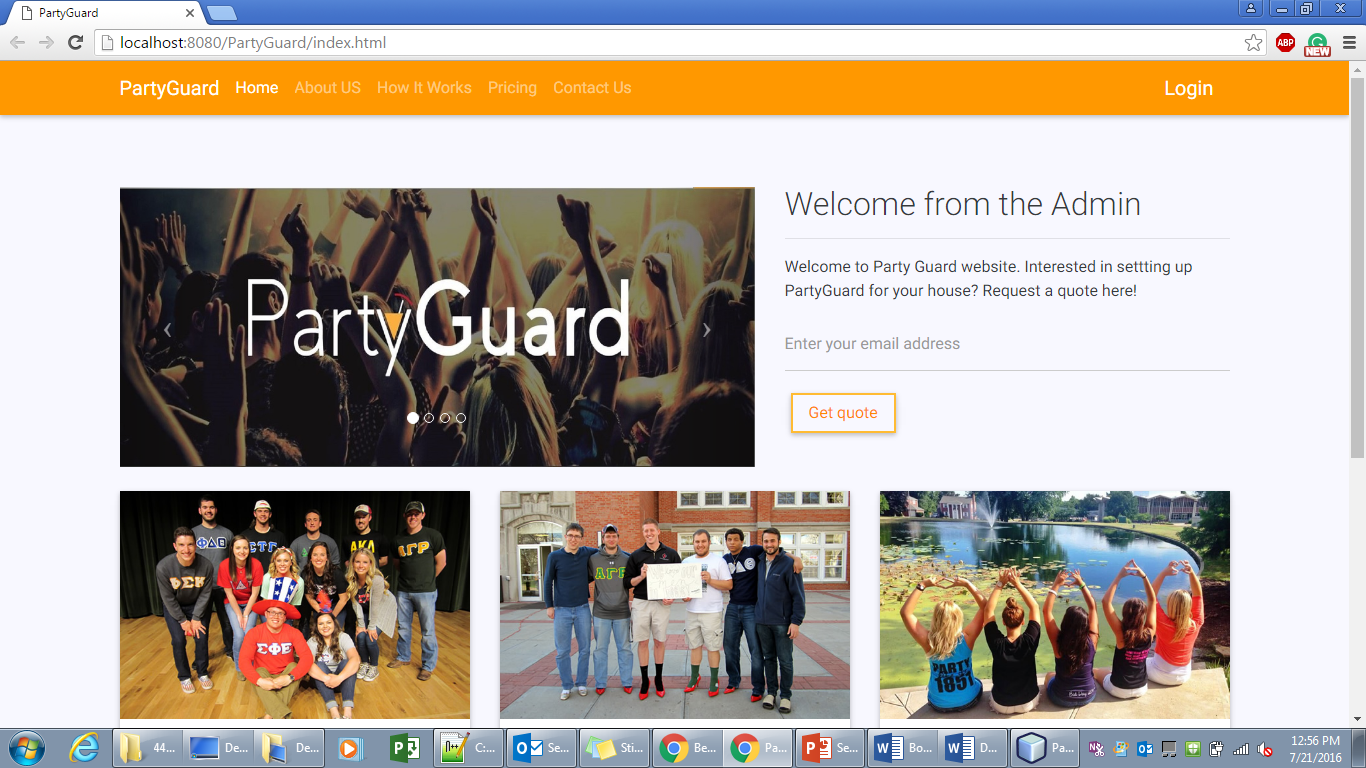
## 3.5 Communications Interfaces

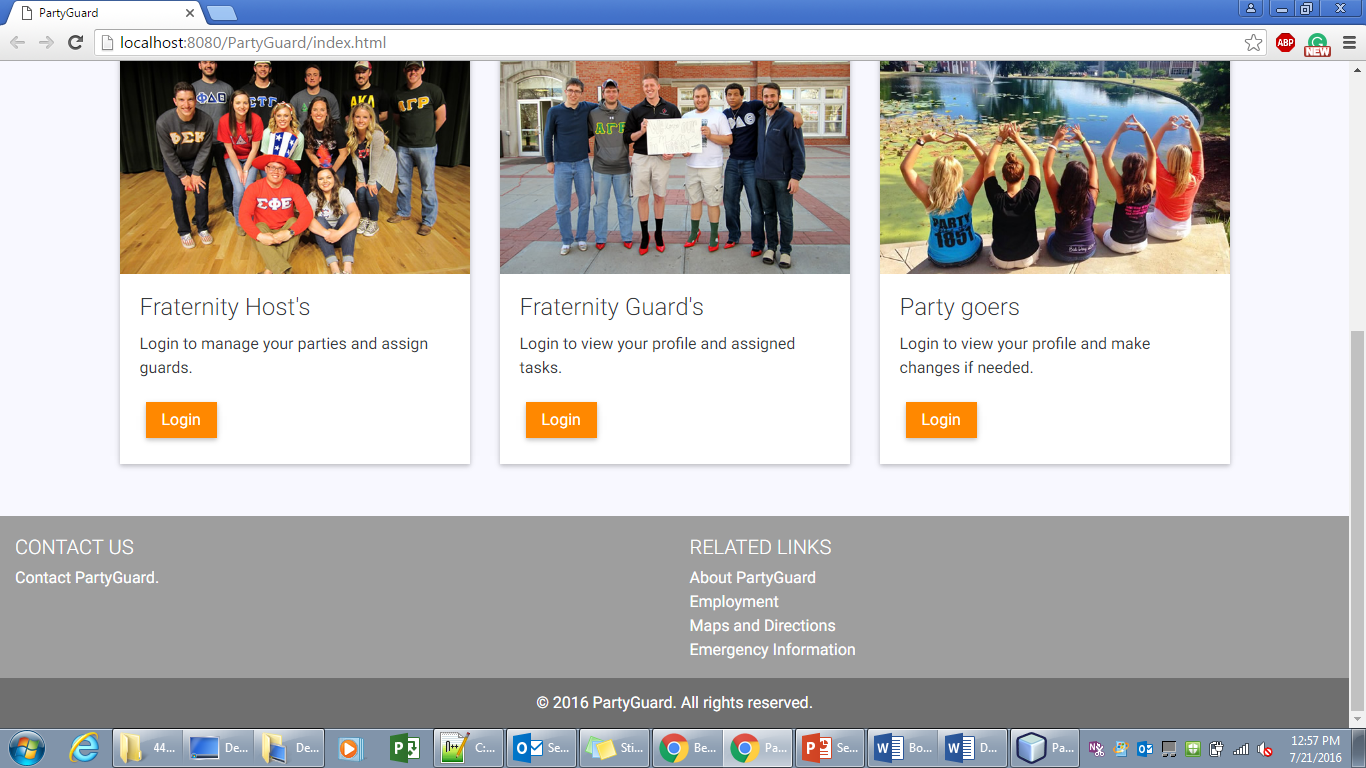
Communication in the application is mainly done by email, also we are planning to use mobile text messages to pass some quick information. The communication standard that we will be using is HTTP.

# 4. Initial UI design screens:

**Home Page:**

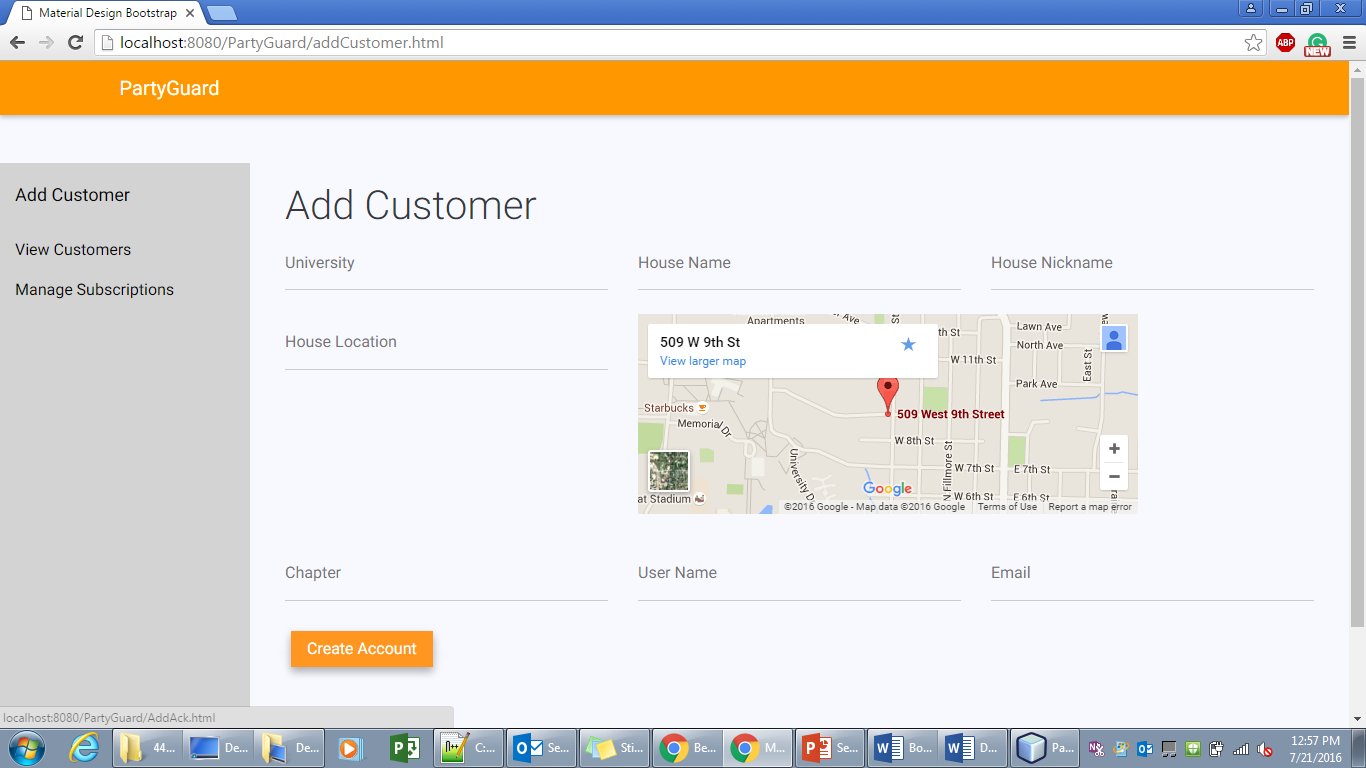
* New user can enter his email address and request for quote.
* Master user can navigate to Login page and login to add customers.
* We have 3 cards (Host, Guard, and Member) which has individual Login buttons.
* Host User can either login by clicking ‘Login button’ on navbar or he can click ‘Login’ button available on his card, as show below.
* Similarly Guard/Member can login using either of the two options provided.



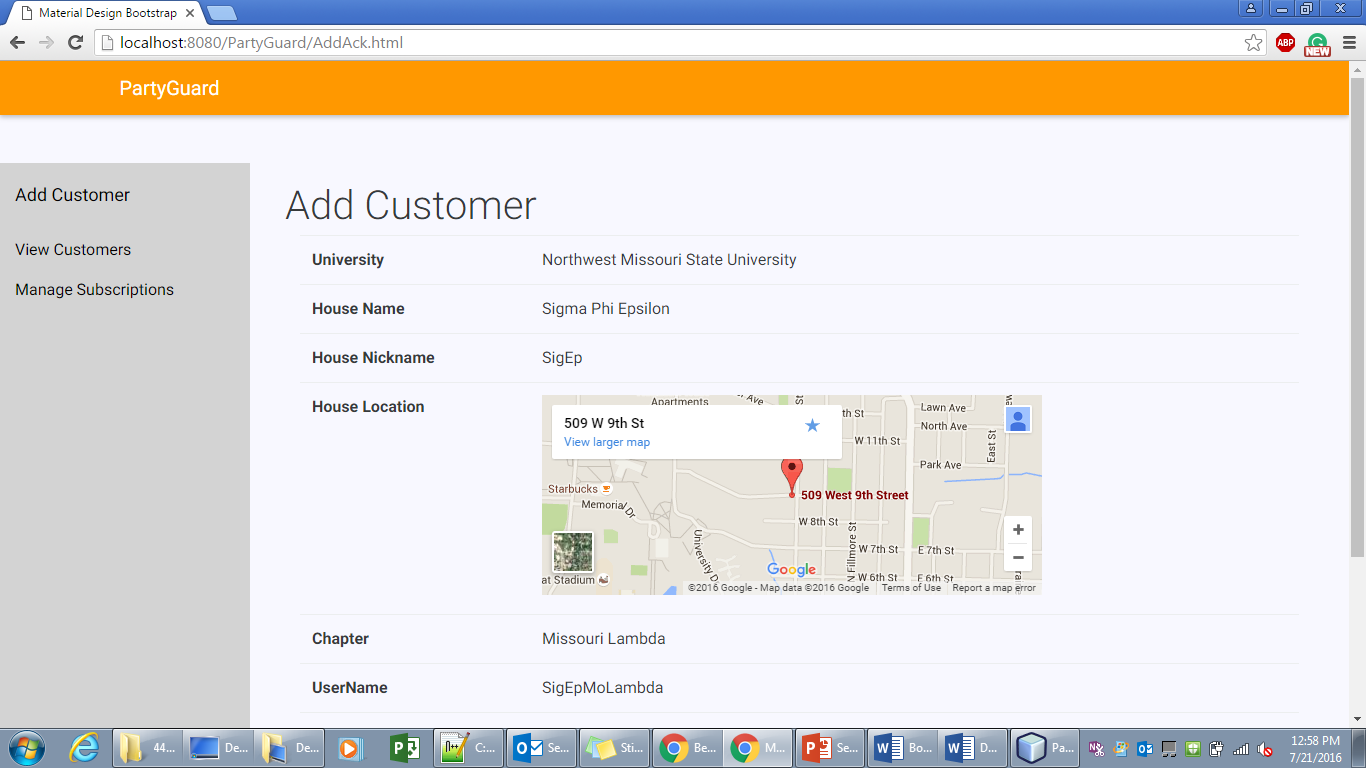


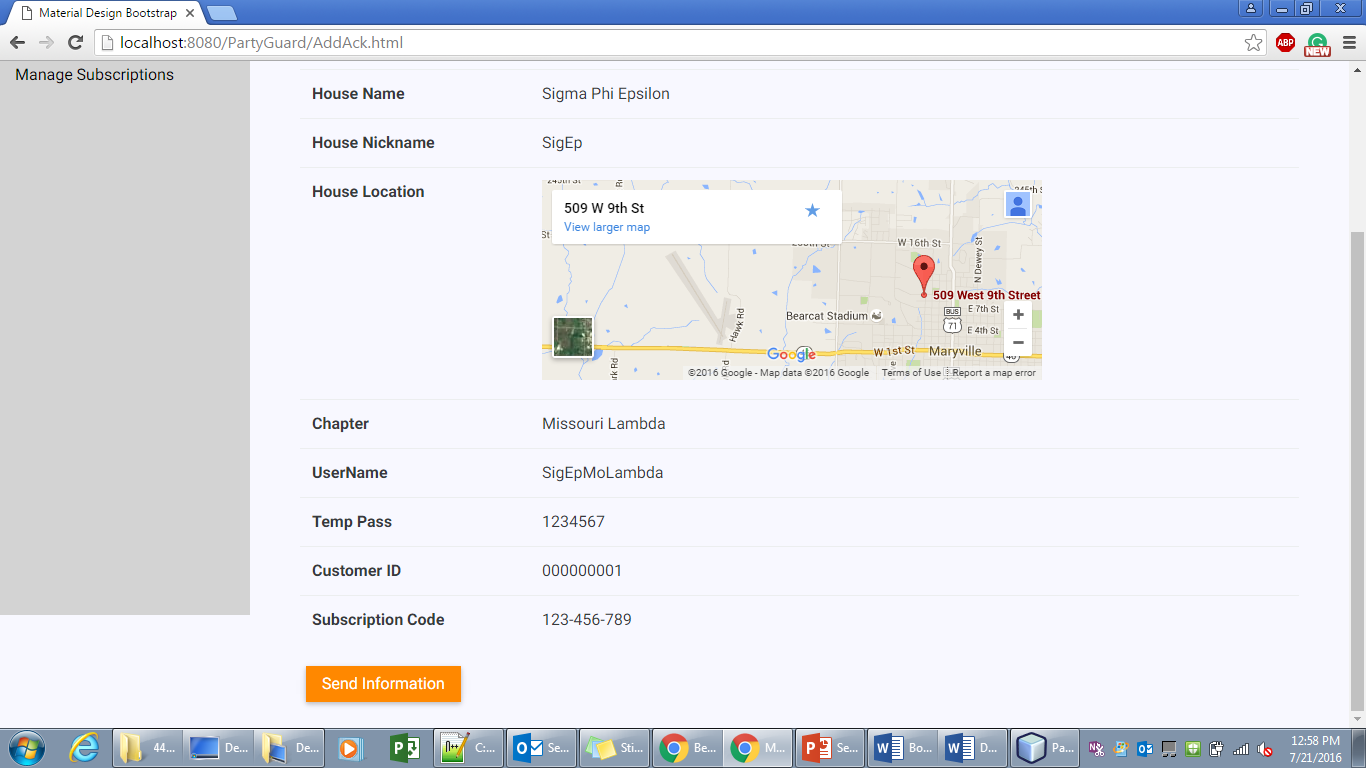
**Add Customer page:**

* Master user can add new customer and create new account for them.



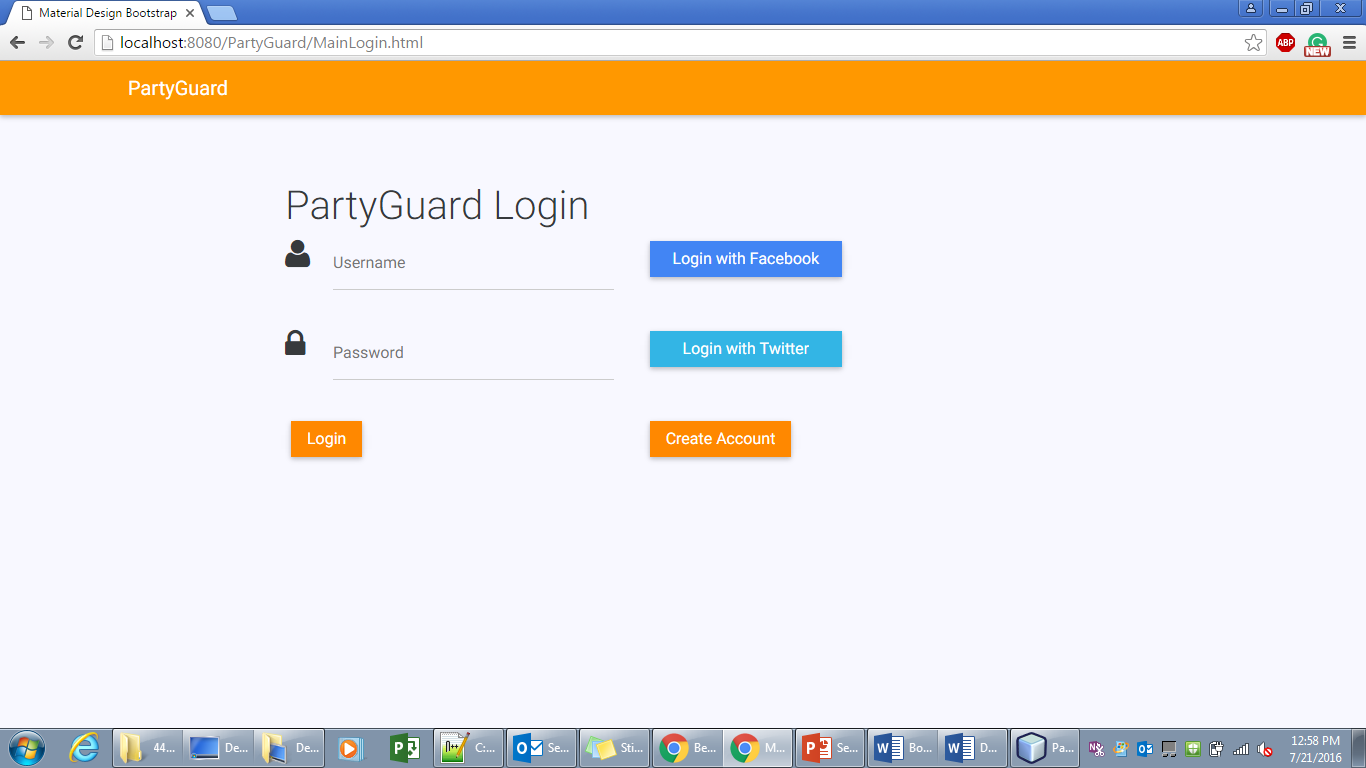
* Once the account has been created, a confirmation page with Temp password, Customer ID and Subscription code will be displayed.





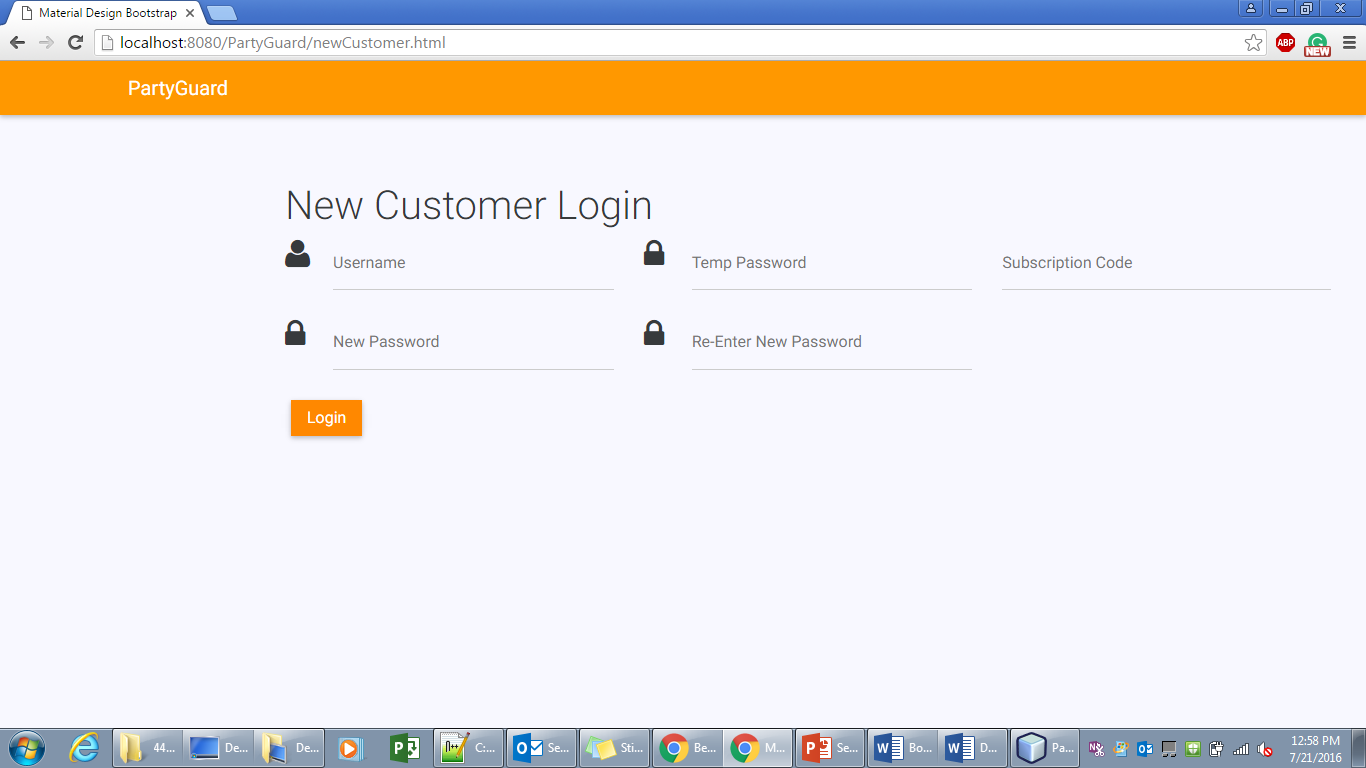
**Login page:**

* User can either login using their PartyGuard username or using social network logins.

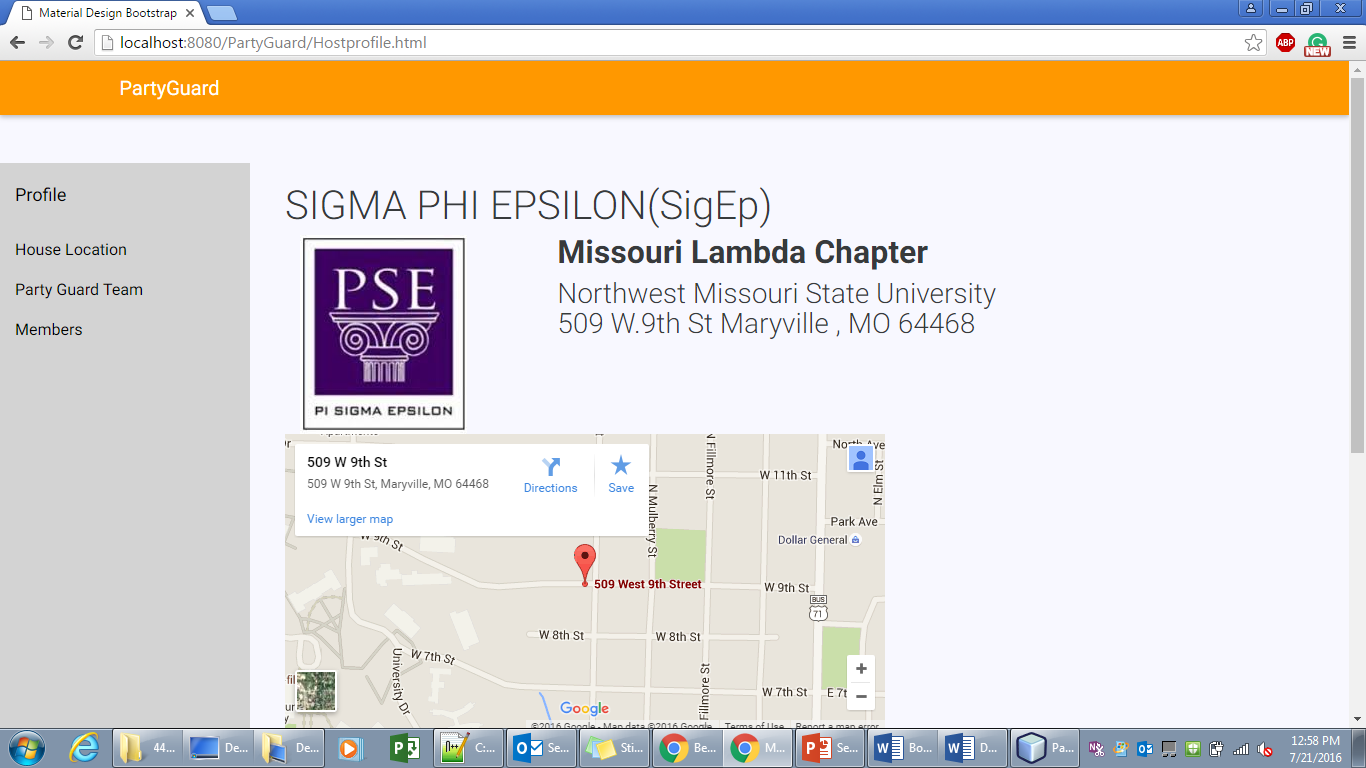


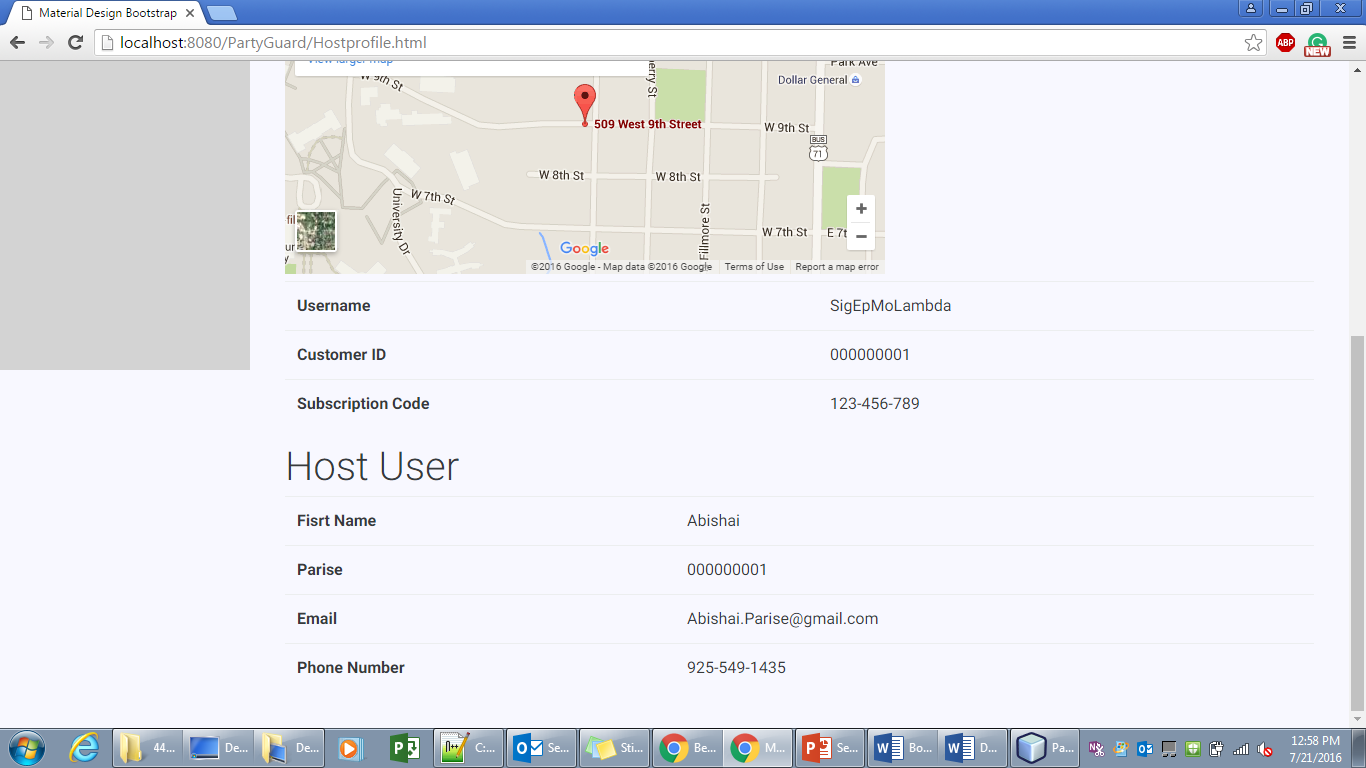
**Host User login:**

* Once the host user logins into his account for the first time, He should provide the temporary password and subscription code in order to access his profile.

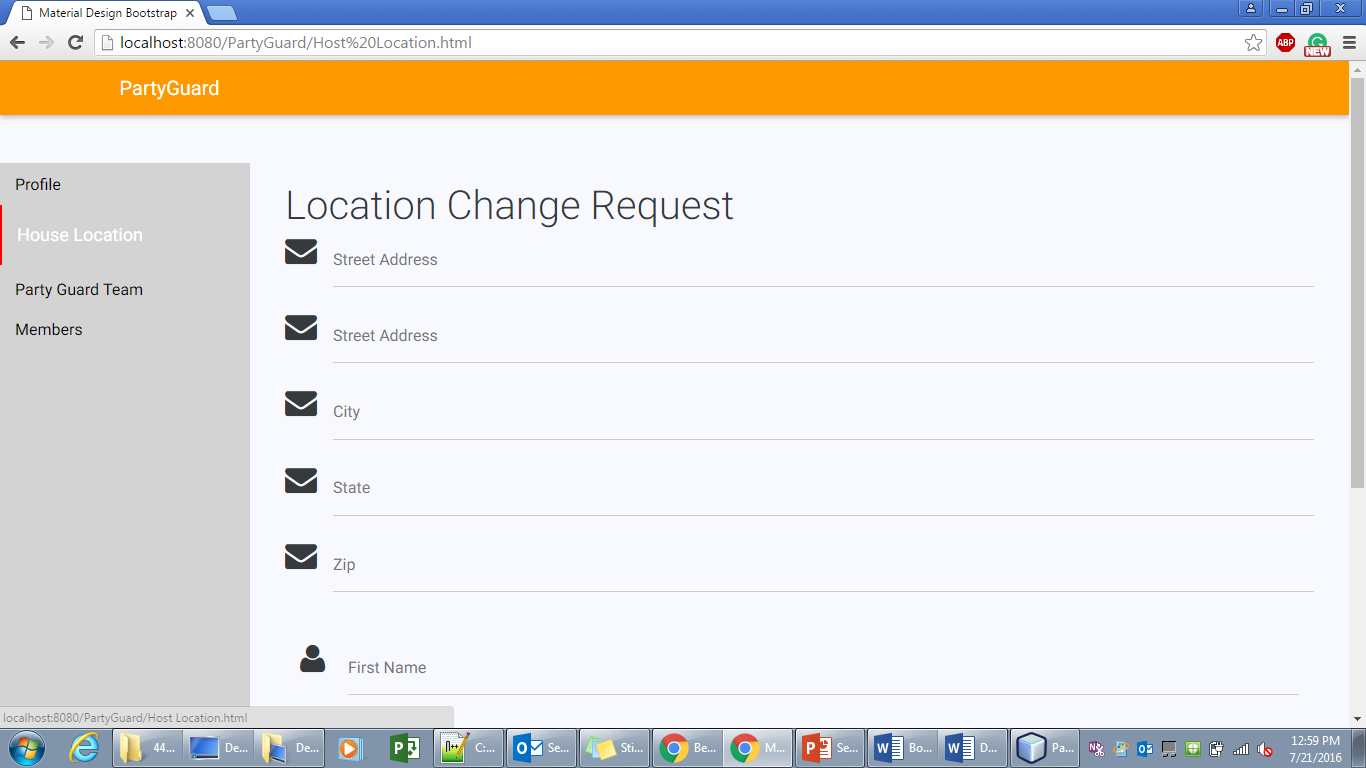


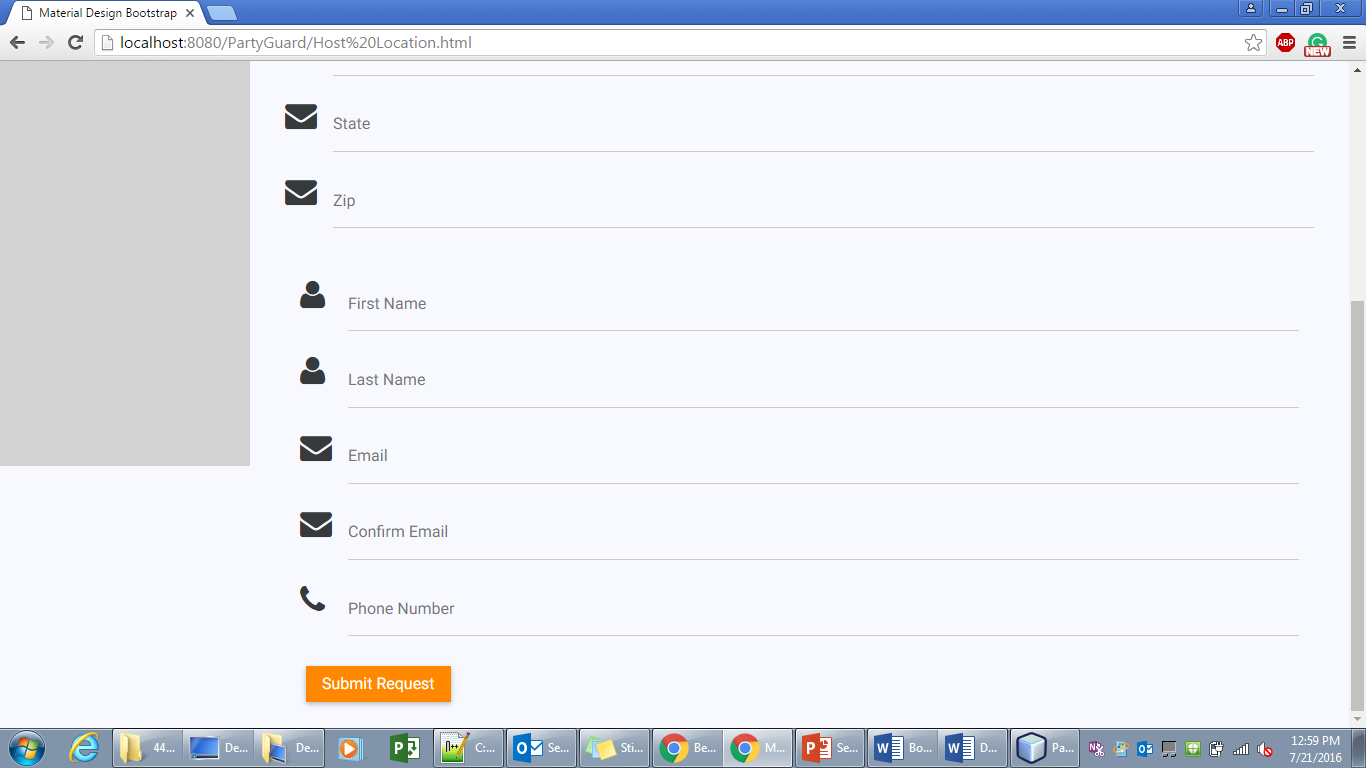
**Host profile page:**





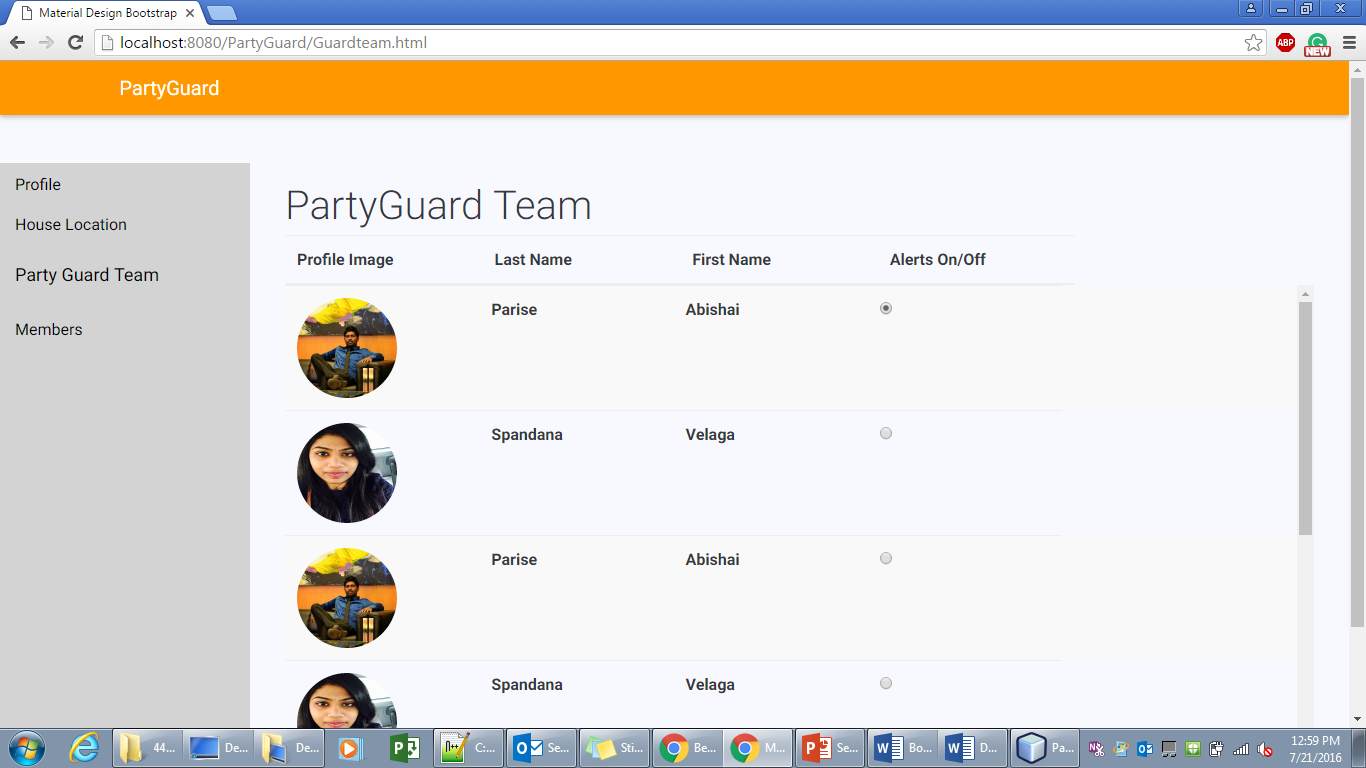
**Location Change Request page:**



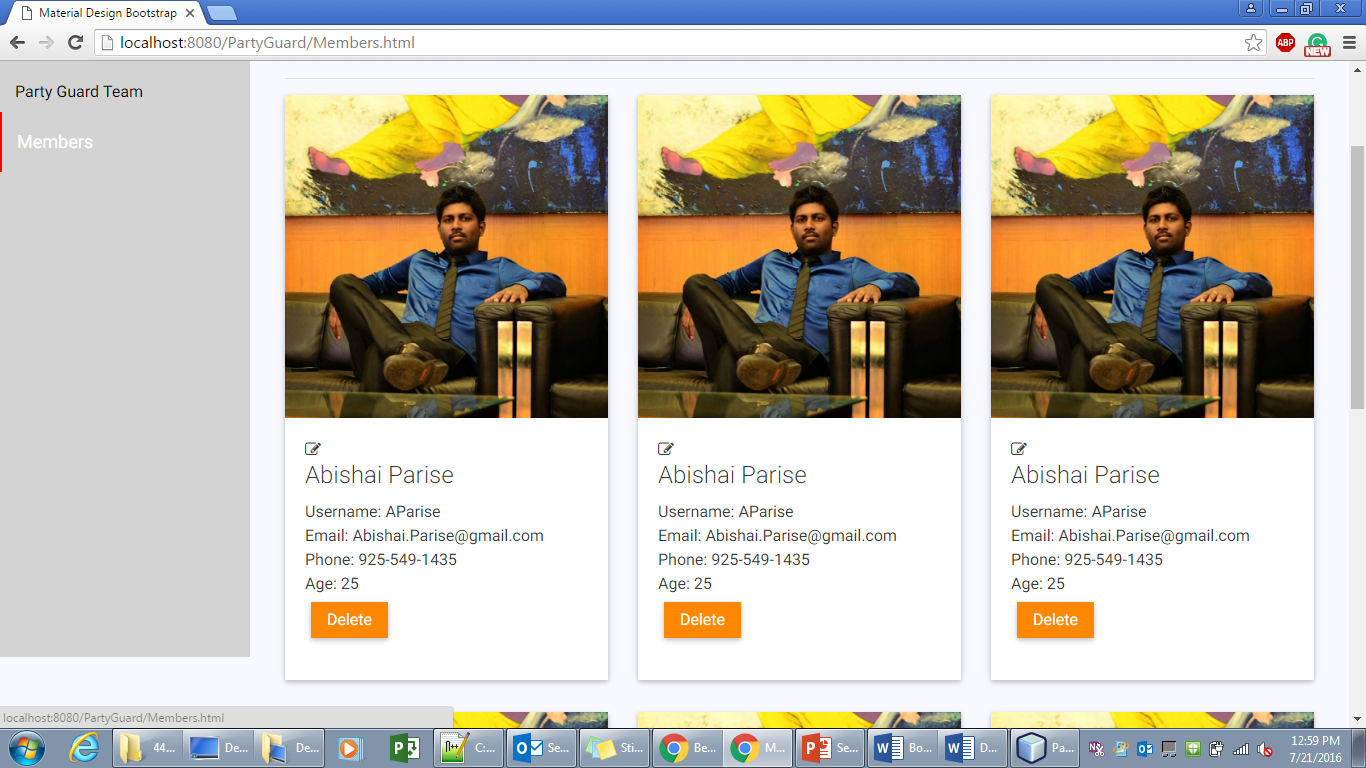


**PartyGuard Team page:**

* List of all the members will be displayed here. Host can select Guards for the party in this list.



**Member’s page:**



**Guard/Basic member profile page:**

