Software Requirements Specification

for

Apart-maint

Version 1.0 approved

Prepared by: Sriram Arvind

Organization: Cosmo

Date: 18/09/2016

Table of Contents

Table of Contents [ii](#__RefHeading___Toc441230970)

Revision History [ii](#__RefHeading___Toc441230971)

1. Introduction [1](#__RefHeading___Toc441230972)

1.1 Purpose [1](#__RefHeading___Toc441230973)

1.2 Document Conventions [1](#__RefHeading___Toc441230974)

1.3 Intended Audience and Reading Suggestions [1](#__RefHeading___Toc441230975)

1.4 Product Scope [1](#__RefHeading___Toc441230976)

1.5 References [2](#__RefHeading___Toc441230977)

2. Overall Description [2](#__RefHeading___Toc441230978)

2.1 Product Perspective [2](#__RefHeading___Toc441230979)

2.2 Product Functions [2](#__RefHeading___Toc441230980)

2.3 User Classes and Characteristics [3](#__RefHeading___Toc441230981)

2.4 Operating Environment [4](#__RefHeading___Toc441230982)

2.5 Design and Implementation Constraints [4](#__RefHeading___Toc441230983)

2.6 User Documentation [4](#__RefHeading___Toc441230984)

2.7 Assumptions and Dependencies [4](#__RefHeading___Toc441230985)

3. External Interface Requirements [4](#__RefHeading___Toc441230986)

3.1 User Interfaces [4](#__RefHeading___Toc441230987)

3.2 Hardware Interfaces [4](#__RefHeading___Toc441230988)

3.3 Software Interfaces [5](#__RefHeading___Toc441230989)

3.4 Communications Interfaces [5](#__RefHeading___Toc441230990)

4. System Features [5](#__RefHeading___Toc441230991)

4.1 System Feature 1 [5](#__RefHeading___Toc441230992)

4.2 System Feature 2 [6](#__RefHeading___Toc441230993)

4.3 System Feature 3 [7](#__RefHeading___Toc441230993)

4.4 System Feature 4 [7](#__RefHeading___Toc441230993)

4.5 System Feature 5 [8](#__RefHeading___Toc441230993)

4.6 System Feature 6 9

5. Other Non-functional Requirements [10](#__RefHeading___Toc441230994)

5.1 Performance Requirements [10](#__RefHeading___Toc441230995)

5.2 Safety Requirements [10](#__RefHeading___Toc441230996)

5.3 Security Requirements 10

5.4 Software Quality Attributes 10

5.5 Business Rules [10](#__RefHeading___Toc441230999)

6. Other Requirements [11](#__RefHeading___Toc441231000)

Appendix A: Glossary 11

Appendix B: Analysis Models 11

Appendix C: To Be Determined List 11

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The web/mobile application being developed provides a digital environment to the people living in the same apartment or community and helps them to communicate and interact better with the authorities/management. It also provides the authorities an opportunity where they can perform tasks like providing feedback and respond to complaints by the house owners. This application basically automatizes and digitize the function of all the regular society procedures, hence working as a virtual place where users (tenants) can get updates and alerts on what is going around in the society and information about the upcoming events.

## Document Conventions

This following System Requirement Specification follows the standard text conventions where the main heading is of font size 16 and the text is of size 11 with Arial font. The document also follows the standard Convention for numbering the contents of the text.

PHP - Pre Hypertext Processor.

HTML - Hypertext Markup Language.

Code-Igniter – A PHP based framework.

SQL- Structure Query Language

MongoDB- Database software

Microsoft Azure- Cloud database

## Intended Audience and Reading Suggestions

This document is intended for the existing customers and all the potential clients’ .This SRS explains all the parameters of the product and helps in analyzing the same.

## Product Scope

This Product is intended for the purpose of automation of the apartment or gcommunity and its working. The product will be used by home-owners and authorities to connect to each other on a virtual platform to save time and increase the efficiency of the current existing system which is manual in nature.

## References

<https://azure.microsoft.com/en-in/services/app-service/web/>

<https://www.codeigniter.com/docs>

<https://docs.mongodb.com/?_ga=1.50397930.270198069.1474191331>

# Overall Description

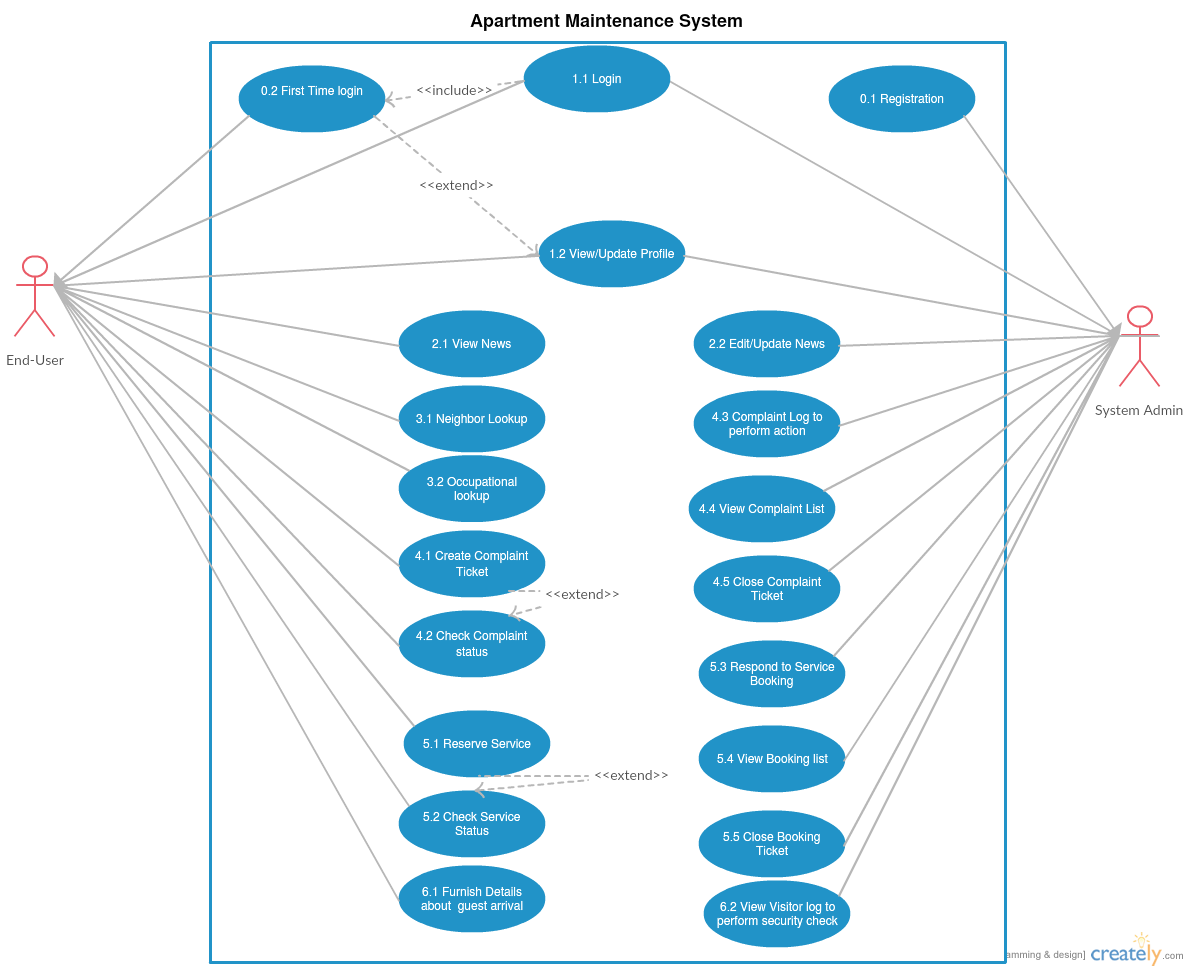
## Product Perspective

This web/mobile application is a project which has one objective in mind and that is to create an efficient virtual working environment for the home owners and tenants alike. The user from same society can share and see what and where are all the events happening all around the neighborhood and also be updated about their surroundings.

## Product Functions

1. Service feature allows you to book electrician, plumber etc. and track the progress of the same online.
2. Complaint functionality solves all the issues arising in and around the apartment environment.
3. News System allows the tenant to be updated about the society events and all the news around their community.
4. Visitor system to improvise general security.

## User Classes and Characteristics



## Operating Environment

The software is based on hybrid system that makes it the sole reason of making the architect cloud based. Cloud system is based on Microsoft Windows Azure. Web technology is based on PHP (5.3.8 or 7.0.8) and is implemented using framework Code-Igniter (3.1.0) and Slim 3 (for APIs). Database technology is based on MySQL and MongoDB.

## Design and Implementation Constraints

Good Internet is the backbone for this project. Limitations of the project are modification of the web app for some specific place which could lead to non-optimized functionality and hamper the functioning of the app. Maintainability can be obtained by onsite or offsite engineers so one can talk to management and work on a problem from other place without actually being present there.

## Assumptions and Dependencies

This software depends on Azure cloud service by Microsoft which is a third-party service. Some functions may not work if this service is not available. For payment and access management it should be noted that third-party API's are used any bug in those API or any unstable update on those API's may damage the software and it working and may impair some of its functionality which could result in some down times. In some restricted environment there can be problems like slow internet or unavailability of service such as Azure on that network which could result in less optimal functioning to no functioning of the web app.

# External Interface Requirements

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## Hardware Interfaces

Supported device list for Apart-maint web application are any smart devices with a non-restricted internet connection this includes smart-phone, tablets etc. For Administrator side a Computer with a non-restricted internet connection and Good processing power and LAN is recommended. A Computer with Intel i5 or AMD A10+ processors and a basic graphic processor by NVIDIA or AMD is recommended. A router with good bandwidth compatibility 4+ LAN port support and cat 6 LAN cables are recommended. To access the web-app the internet is sufficient for Residents. However for easy working of the app on admin side, the recommended hardware should be used.

## Software Interfaces

The web-app uses basic web browser which is available in any smart-device now a days. Also there is no restriction in the operating system as the app is online and readily available. Azure web service used for hosting the app is an open service which is available on any non-restricted internet. Interface Protocols for Azure service can be found at: [Azure Web Service](https://azure.microsoft.com/en-in/services/app-service/web/)

## Communications Interfaces

Web browser with HTML5 support and JavaScript support are needed which are there in most of the new smart-devices these days. Availability of internet with HTTP and HTTPS and without restrictions is a must. Administrator side should have a gigabit LAN communication for seamless communication.

# System Features

## Login

4.1.1 Description and Priority

A feature whose main function is to determine the user profile.

4.1.2 Stimulus/Response Sequences

The user will provide the mobile number as the username and the password so that the credentials can be checked.

If the user is logging in for the first time, an OTP is generated so that the user can login safely and then can also change the password and the user is also directed to update this profile with all the necessary data fields. Once the user had updated his profile page he is redirected to the main user dashboard.

If the authentication fails, an system error is displayed on the login page stating “invalid credentials “ and the user forgets his credentials is given a chance to recover them.

There are two phases for this module, one is the user phase and other is the admin phase.

Priority: High

4.1.3 Functional Requirements

|  |  |  |
| --- | --- | --- |
| Requirement Identifier | Name | Description |
| LF\_1 | Login Feature 1 | Check for username input box, display error if input is anything other than 10 digit number. |
| LF \_2 | Login Feature 2 | Then check for password input box, display error if input is anything other than alphabet. |
| LF \_3 | Login Feature 3 | Input Data is transferred to controller in POST method. |
| LF \_4 | Login Feature 4 | Check for the Phase i.e. User Phase and Admin Phase. |
| LF \_5 | Login Feature 5 | First time Login Check |
| LF \_6 | Login Feature 6 | Update new password and Update Profile methods (if LF\_5 is executed) |
| LF \_7 | Login Feature 7 | Display the Main User Dashboard (if LF\_4 is user Phase) |
| LF \_8 | Login Feature 8 | Display the Main Admin Dashboard (If LF\_4 is admin phase) |
| LF \_9 | Login Feature 9 | Register all the apartment users |

## News System

4.2.1 Description and Priority

A News Alert feature to update about all the events and news of the apartment or community.

Priority: Low

4.2.2 Stimulus/Response Sequences

There are two phases of this feature, one is the user phase and other is the admin phase.

The user (tenant) when is logged in to the application, a menu is displayed which shows the list of all happenings of the apartment or community in form of news.

When the administrator is logged into the website, a menu named News is displayed which shows three sub menu’s:

* View News
* Edit/Delete News
* Add News

4.2.3 Functional Requirements

|  |  |  |
| --- | --- | --- |
| Requirement Identifier | Name | Description |
| NSF\_1 | News System Feature 1 | This phase checks if user is logged in or not, if yes then check the phase. |
| NSF \_2 | News System Feature 2 | If the phase is user phase, load the News System of user phase. |
| NSF \_3 | News System Feature 3 | If admin phase, load the News System of admin phase. |

## Look UP

4.3.1 Description and Priority

A neighbor information system, where basic information about nearby house is known.

Priority: Medium

4.3.2 Stimulus/Response Sequences

There is only one phase, i.e. user phase.

The tenant can check who are their neighbors in the building depending on the structure of apartment and lso the number of houses.

A List of the nearby house is generated which is read only type

4.3.3 Functional Requirements

|  |  |  |
| --- | --- | --- |
| Requirement Identifiers | Name | Description |
| NLF\_1 | Neighbor Lookup 1 | Check if user is logged in or not, if yes then check the phase |
| NLF \_2 | Neighbor Lookup 2 | If user phase, load the Neighbor system depending on the house number. |
| NLF \_3 | Neighbor Lookup 3 | A list generator of nearby house which is only read-only. |

.

## Complaint System

4.4.1 Description and Priority

This feature is a complaint booking system where the user (tenant) can book complaints such as lift breakdown, parking space problem etc. and can also track the progress of the complaint. This also allows the administrators to update the user with the progress such as course of action hence allowing the user to track if the complaint has been resolved or not.

Priority: Medium

4.4.2 Stimulus/Response Sequences

There are two phases i.e. user phase and admin phase.

The user phase can book a complaint and generate a booking ticket number which is used to track the complaint status.

The admin phase can view the complaint by all the tenants and can update the ticket number so that the tenant can know that the complaint has been resolved.

4.4.3 Functional Requirements

|  |  |  |
| --- | --- | --- |
| Requirement Identifiers | Name | Description |
| CS\_1 | Complaint Service 1 | Check if user is logged in or not, if yes then we check the phase |
| CS\_2 | Complaint Service 2 | If user phase, load the complaint booking method |
| CS\_3 | Complaint Service 3 | Check the input data from the user such as complaint type and complaint description. |
| CS\_4 | Complaint Service 4 | If admin phase, load the complaint status method |
| CS\_5 | Complaint Service 5 | Check the list of booking tickets generated for the moment and allot necessary employee for the requested tickets |
| CS\_6 | Complaint Service 6 | Update the ticket number so that the user can track their complaint request |

## Service Booking

4.5.1 Description and Priority

A system that helps in booking a service. This feature allows the user (tenant) to book services such as plumber, carpenter, housemaid etc. and the tracking system allows administrators to update the user with the progress of the work.

Priority: High

4.5.2 Stimulus/Response Sequences

There are two phases i.e. user phase and admin phase.

The user (tenant) can book a service and generate a service booking ticket number, which can be used to track the booked service. This service can also be cancelled.

The administrator can view the booked service by the tenants and allot the right person for the service. The administrator can update the booking service number so that the tenant can know what the progress of the work etc. is.

4.5.3 Functional Requirements

|  |  |  |
| --- | --- | --- |
| Requirement ID | Name | Description |
| SS\_1 | Service System\_1 | First we check if user is logged in or not, if yes then we check the phase |
| SS\_2 | Service System\_2 | If user phase, load the service booking method |
| SS\_3 | Service System\_3 | Check the input data from the user such as service type, work description and estimated amount of work and availability time. |
| SS\_4 | Service System\_4 | If admin phase, load the service tracking method |
| SS\_5 | Service System\_5 | Check the list of service booking tickets generated and allot necessary employee for the work. |
| SS\_6 | Service System\_6 | Update the service booking ticket number so that the user can track their service request. |

## Visitor system

4.6.1 Description and Priority

A feature that updates about any arrival of visitor to the security officials for better security of the apartment tenants.

Priority: Low

4.6.2 Stimulus/Response Sequences

There are two phases of this feature, one is the user phase.

The user phase (tenant) when is logged in to the website, a visitor menu will be displayed to them having all the information regarding the details of visitors including their expected time of arrival etc.

4.6.3 Functional Requirements

|  |  |  |
| --- | --- | --- |
| Requirement ID | Name | Description |
| VF\_1 | Visitor Feature 1 | Check if user is logged in or not, if yes then check the phase. |
| VF\_2 | Visitor Feature 2 | if user phase, load the visitor updated menu |
| VF\_3 | Visitor Feature 3 | If admin phase, load the list of visitors for the day. |

# Other Nonfunctional Requirements

|  |  |  |
| --- | --- | --- |
| Requirement ID | Name | Description |
| 5.1 | Performance Requirements | The ideal environment for the working of the app is where all the tenants and home owners can work together on a real time system. |
| 5.2 | Safety Requirements | There are no safety requirements as such |
| 5.3 | Security Requirements | Loss of the data, corruption with the data are the privacy issues and security concerns with this app. Using Azure platform we regulate the security concerns to the application. |
| 5.4 | Software Requirements | Availability- Only requires access to Azure Web Services  Maintainability- Easy to maintain its services  Re-usability- Can be used by each and every apartment anywhere in the country |
|  |  |  |

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>