<u>SOFTWARE REQUIREMENTS SPECIFICATION</u>

for

HoGo - An Apartment Community App

All ok by advaith

Version 1.0 approved

Prepared by: Team members:

Shriya Harish (PES2UG20CS463)

Aarav Babu (PES2UG20CS486)

Aditya Lawankar (PES2UG20CS488)

Advaith Shet (PES2UG20CS490)

Of PES University

Created on 04/09/2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

Introduction

Purpose

The purpose of this document is to present a detailed description of the application. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

Intended Audience

This document is primarily intended for the:

- Developers of this software

- Software engineers who would work on further development of the project

- The professors who would review the document and finally,

- Clients, that is, residents of an apartment.

Product Scope

The product being developed is a society management software for apartment residents.
The product will be designed to simplify things for a resident by

- Manage security from any place through verification of guests/domestic help/delivery executives requesting access to your home.
- Pay and maintain your maintenance bills.
- Residents can chat with their neighbours, discuss/organise events and raise complaints that will directly inform the authorities.
- Provides extension information about the flats and common areas such as main gate security, maintenance staff, club house

The product would hence make house management convenient and hassle free by laying off the need to rely on others which is time consuming and sometimes undependable.

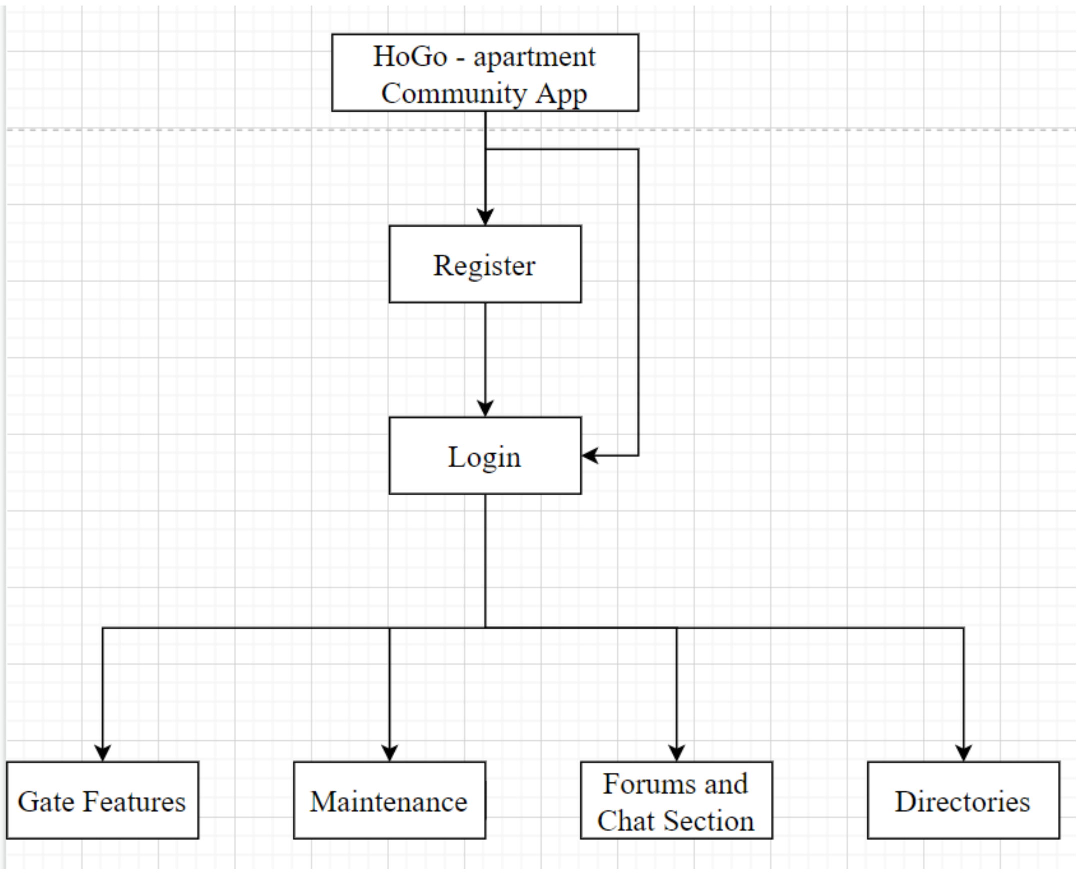
Overall Description

Product Perspective

HoGo is a new, self-contained apartment community app aimed at providing a platform for the residents to communicate, manage their maintenance bills, raise issues and its core feature: providing security by verifying identities of people entering the apartment that includes guests, domestic help and package deliveries. This application makes it hassle free for residents to manage security better as they can accept/deny entry of others from any place. Complaints can be placed at your fingertips with convenience. Communication with your neighbours made easy. It is the one stop solution for all your apartment-related needs.

Product Functions

- **Gate Features:** A security system that uses QR codes to allow House helps, Delivery Boys, and Guests to enter the community securely. This allows the user to be alerted who is trying to enter the apartment.
- **Customer Registration:** Allows new residents to register and virtually become a part of the apartment community
- **Maintenance**: Pay maintenance bills through the app.
- **Forums and Chat Section:** Residents can chat with their neighbours, discuss/organise events and raise complaints that will directly inform the authorities.
- **Directories:** Provides extension information about the flats and common areas such as main gate security, maintenance staff, club house etc



User Classes and Characteristics

The system will support two kinds of user privileges:

- Apartment residents
- Apartment managers and council members

Sl. No	Type of user	Usage
1.	Apartment residents	Access gate features for visitor alerts, forums and chat section for apartment-wide communication, maintenance to pay bills, and directories to access extension information. Registration section can be used to set up account in the database
2.	Apartment managers and council members	Access forums and chat section for communication purposes, maintenance to track resident bill payments and directories to update extension information. Registration section and database can be used to access residents' apartment information

Operating Environment

The software will be designed to work on any version of Windows, Linux (kernel 2.7 and above) and Mac platform. The software is completely web based and runs on popular web browsers namely firefox, chrome, internet explorer (IE8 and above). These web browsers are preferred since they support HTML.

Design and Implementation Constraints

User needs a standard web browser to support MERN Stack and good network connection to support payment and real-time chat and visitor notification features.

Assumptions and Dependencies

The user is familiar with internet and web based software like social networking sites. The browsers which the user is using either Google Chrome 10.0 and above or Mozilla Firefox 4.0 and above.

External Interface Requirements

User Interfaces

The user interface will be in the form of a website where users interact through clicks. The home page will display all the available services. For the visitors entering the apartment, the user has to either accept or deny their access on the app. To communicate with neighbours or the management authorities, a user friendly chatbox is available to send and receive texts. The app also provides a payment mechanism for user to clear their maintenance dues at convenience.

Software Interfaces

The services of the product is accessed via a web application that will depend on the following languages :
MongoDB - Database to store information about the residents, their maintenance and chat history.
ExpressJS - Backend tool to route between pages/different services.
ReactJS - Develop the front end part of the webpage with a dynamic touch.
NodeJS - Runtime environment to run all the Javascript for backend services

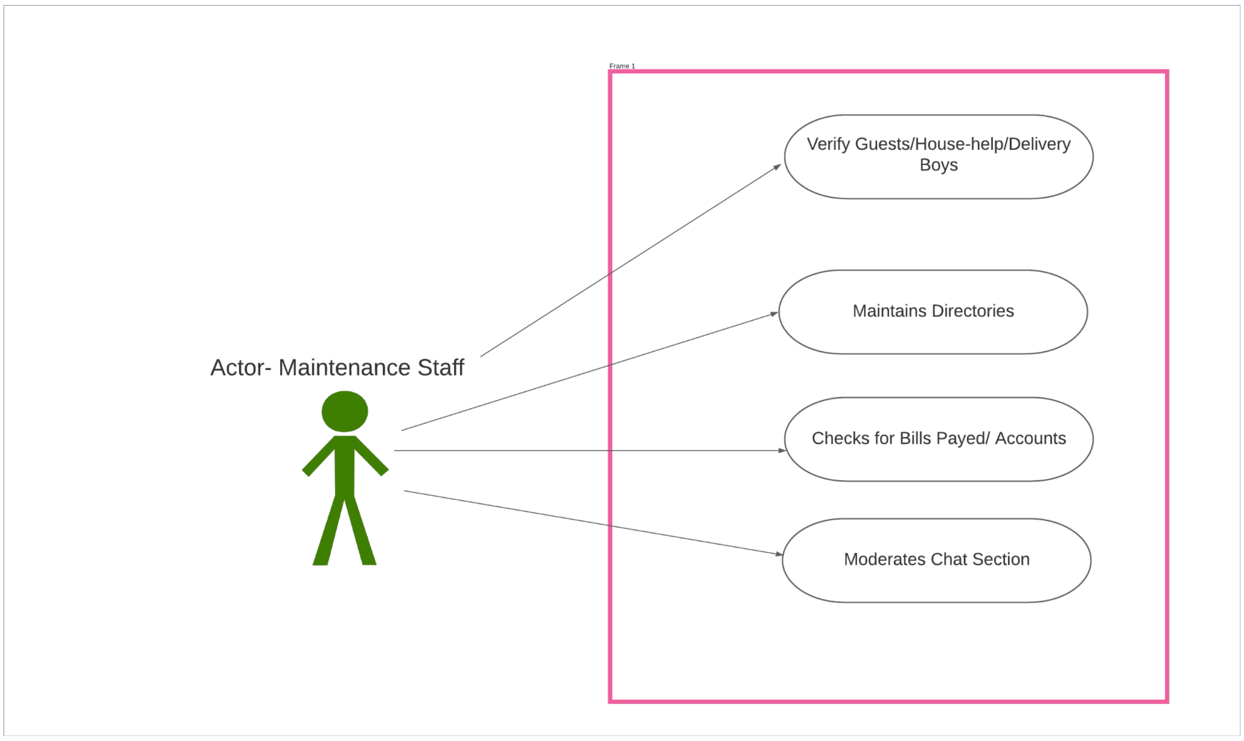
The application is operating system independent, that is it can be viewed/accessed from any operating system.

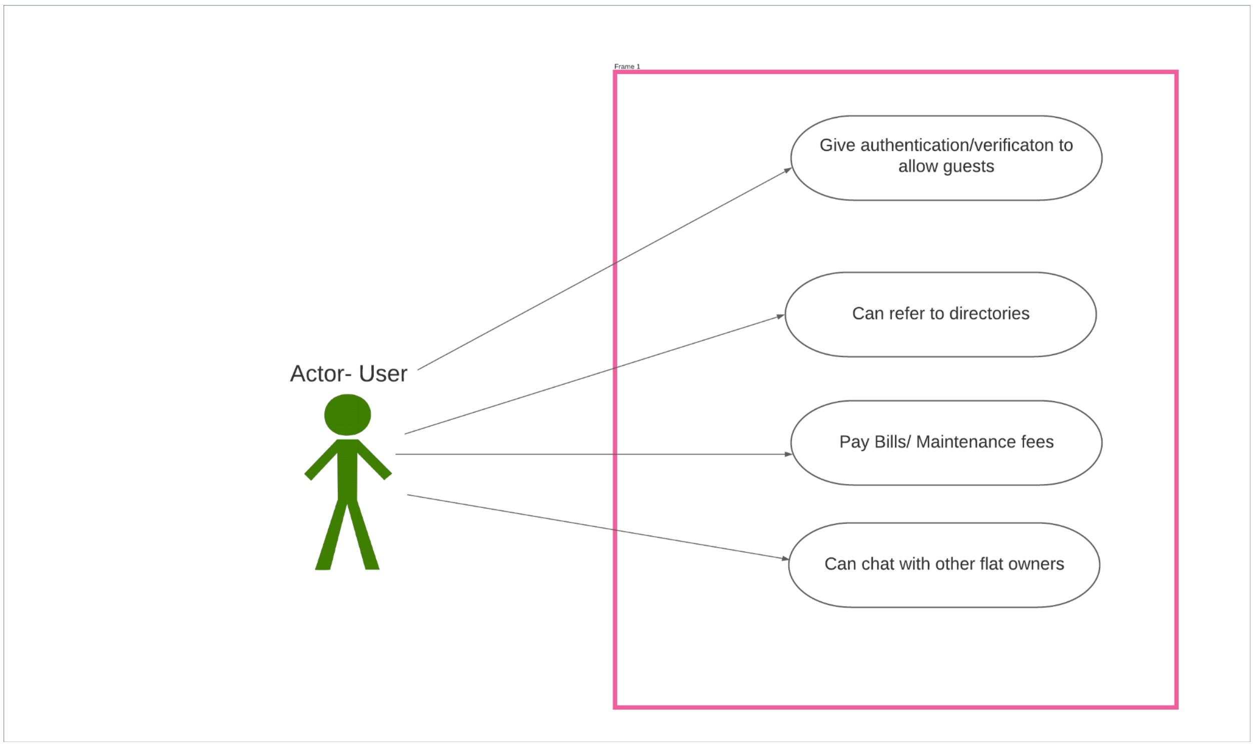
Communications Interfaces

The app will require a suitable web browser that meets the standards of the tech stack used and retrieving and sending data will be done through HTTP requests. There is no additional layer of security apart from the ones provided by the tools used.

Analysis Models

These are the use-case diagrams for the Project:




Frame 1
Actor- User
Give authentication/verificaton to allow guests
Can refer to directories
Pay Bills/ Maintenance fees
Can chat with other flat owners

System Features

This Web-based App is catering to specific users who want a seamless experience to maintain their Apartment/Gated Community. These are the features provided:

Security Features

<u>Description and Priority</u>

This feature allows residents to monitor and provide access on who can enter the apartment. When a guest/Delivery/House Help arrive the Home Owner can give authentication, this allows the security team to ensure the security of the apartment.

<u>Priority</u>: High

<u>Stimulus/Response Sequences</u>

1. Guest arrives.

2. Staff notify resident.

3. User receives notification on app.

4. User gives authorization.

5. Staff allows guest.

<u>Functional Requirements</u>

- Real time notifications.
- When a wrong guest has been indicated by home owner Staff need to be notified.
- TBD

Directories

<u>Description and Priority</u>

Provides extension information about the flats and common areas such as main gate security, maintenance staff, club house etc

<u>Priority</u>: Medium

<u>Stimulus/Response Sequences</u>

1. When a new Home Owner is registered Directory needs to be updated.
2. Any discrepancy in data needs to be modifiable by Staff.

<u>Functional Requirements</u>

- DBMS
- Stored data on a secure platform as it contains confidential information

- Data Abstraction, so other home owners can see only required info.

REQ-1:

REQ-2:

<Dir>

Maintenance

Description and Priority

Provides Home owners a platform to pay maintenance bill and view previous transactions.

Priority: High

Stimulus/Response Sequences

1. View Bill amounts by each home owner.
2. Staff can check status of payments by each owner.
3. Allow users to directly pay online.

Functional Requirements

- Data should be stored on a secured platform.
- Real time updation.

Forums

Description and Priority

Residents can chat with their neighbours, discuss/organise events and raise complaints that will directly inform the authorities.

Priority: Medium

Stimulus/Response Sequences

1.Allow only registered users to chat.

2.Users can chat simultaneously without delay.

3. Users can also ping another resident to have a one on one personal conversation.

Functional Requirements

- Registeres User
- Real time Chat Section

Other Nonfunctional Requirements

Performance Requirements

- Any operation will not take more than 10 seconds to execute.
- Multiple users will be supported.

Safety and Security Requirements

- Anonymous users are not allowed to access this software.
- Only users that have an account and log in can use this software.
- Sensitive payment information is not stored in the server.

Software Quality Attributes

The software will be built using MERN stack which is a popular stack to use for making web applications. MERN stands for **MongoDB**, **Express.js**, **Node.js** and **React.js**.

Using this stack, we can create single page applications which perform better than traditional multipage applications and provide better optimisation and performance for web applications. All technologies that are involved in MERN are open-source.Therefore, there is a lot of support and documentation available for a developer.

Other Requirements

Technical Feasibility

For the implementation of the apartment management software HoGo, the technical resources needed were estimated.

The current solution to the software was decided based on:

- The complexity of the technical resources needed.
- The manpower needed to implement the project.
- Team members prior experience with the technology.
- The stack which provides a complete set of tools required to facilitate agile methodology.