

# Software Design Specification

**for**

**Smart Security**

**Version 3.0**

**Prepared By:**

Tiffany Wong, Helen Hua, Josie Spencer, Aruna Srinivasiah, Connie Diu, Dericka Logan

**Advisor:**

Yuan An

**Stakeholders:**

Tiffany Wong, Helen Hua, Josie Spencer, Aruna Srinivasiah, Connie Diu, Dericka Logan

# Table of Contents

<b>Table of Contents</b>	<b>2</b>
<b>1. Document History</b>	<b>4</b>
<b>2. Introduction</b>	<b>5</b>
<b>Section 2.1: Implementation Platform and Techniques</b>	<b>5</b>
<b>Section 2.2: Features</b>	<b>5</b>
<b>Section 3: UML Class Diagram</b>	<b>6</b>
<b>Section 4: Sequence Diagram</b>	<b>7</b>
Login	7
Send Text Notification	8
Arm/Disarm Alarm	9
Approve Entrant	10
<b>Section 5: State Machine Diagram</b>	<b>11</b>
Home UI	12
Account Creation UI	12
Dashboard UI	12
Approved Entrants	12
Alarm UI	12
Alarm On	12
Alarm Off	12
Alarm Tripped	12
Alarm Triggered	12
Alarm History UI	12
<b>Section 6: Package Diagram</b>	<b>13</b>
Presentation Layer	14
Application Layer	14
Data Layer	14
<b>Section 7: Component Diagram</b>	<b>15</b>
Home Screen UI	16
Alarm UI	16
Approved Entrants UI	16
Alarm Database	16
<b>Section 8: Deployment Diagram</b>	<b>17</b>

<b>Section 9: Implementation Plan</b>	<b>18</b>
<b>Conclusion</b>	<b>18</b>

# 1. Document History

Name	Date	Reason	Version
Tiffany Wong, Helen Hua, Josie Spencer, Aruna Srinivasiah, Connie Diu, Dericka Logan	November 18, 2017	Initial Draft	1.0
Tiffany Wong, Helen Hua, Josie Spencer, Aruna Srinivasiah, Connie Diu, Dericka Logan	February 13, 2018	Fixed Table of Contents, Renamed a few sections, Updated Implementation platform and techniques, Features, Sequence diagrams, and Implementation Plan sections Added previous state machine diagram and revised Google Home integration state machine diagram	2.0
Tiffany Wong, Helen Hua, Josie Spencer, Aruna Srinivasiah, Connie Diu, Dericka Logan	April 4, 2018	Updated state machine diagram section Edited typos/certain terms Updated sequence diagrams, Updated component diagram	3.0

## 2. Introduction

Our team will be working on a senior design project that involves utilizing a smart home device (Google Home and/or Google Home Mini) and turning it into a cheap, multifunctional, and transportable security alarm system. The target demographic would be college students or recent graduates who do not have a lot of disposable income and are renting an apartment. Combining a smart home device with a smartphone will allow users to monitor their home wherever they are.

### Section 2.1: Implementation Platform and Techniques

Implementation platform: Google Home, Web application, Firebase, and DialogFlow

Programming languages: HTML, CSS, JavaScript and Java

Database systems: Firebase

Data set: User information, Transactions

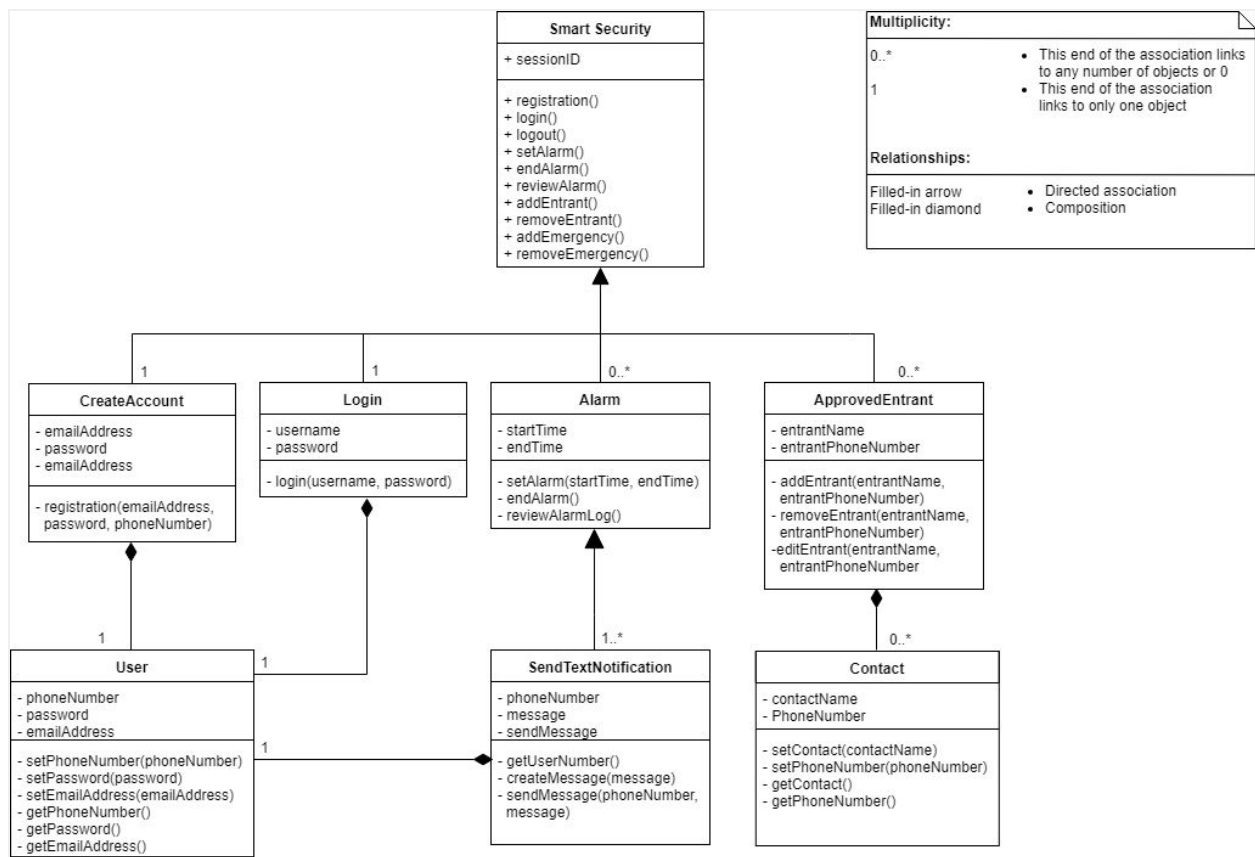
Constraints: Voice recognition, Network connection, Google Assistant

### Section 2.2: Features

Based on our SRS, the highest priority pertains to the web application so the features that were deployed include:

- Login interface
- View alarms interface
- Add and view approved entrants interface

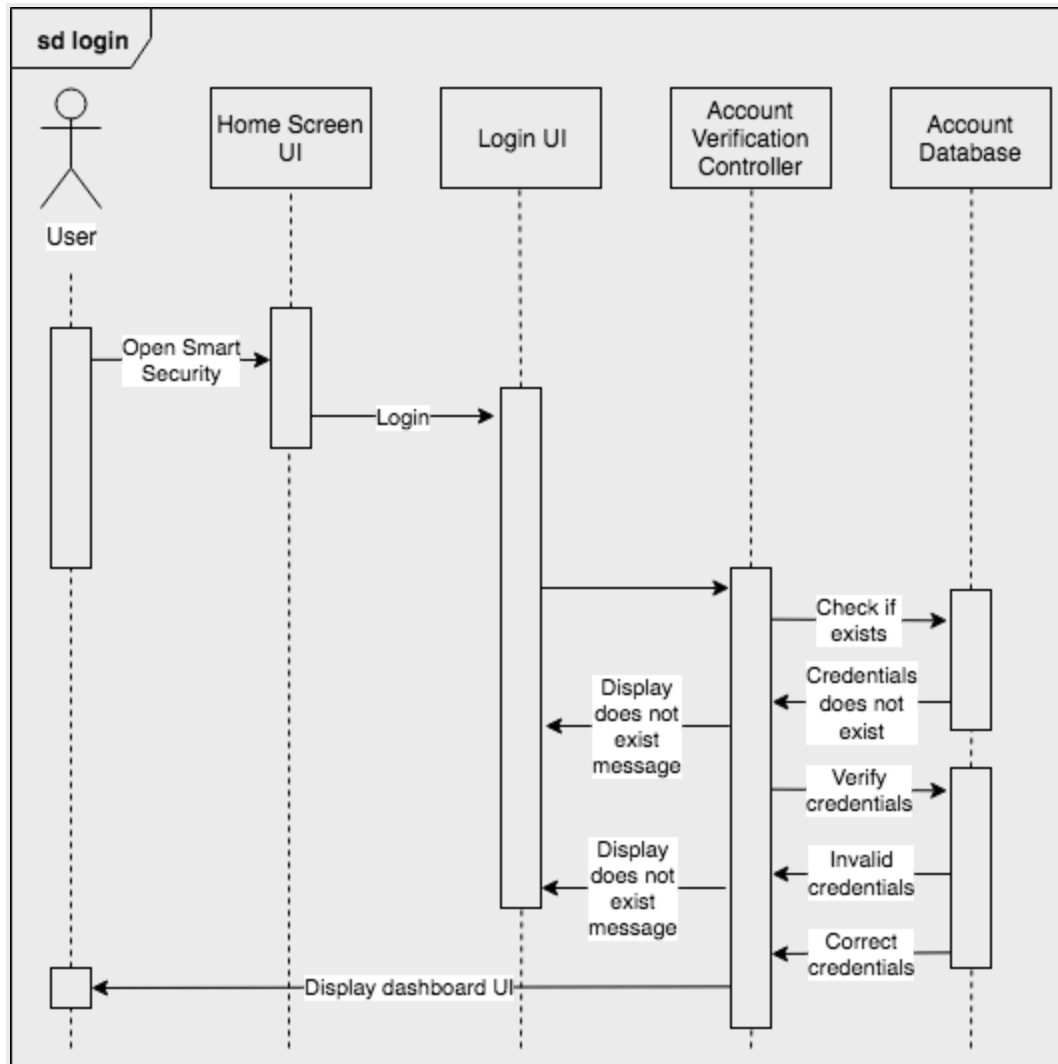
## Section 3: UML Class Diagram



The UML class diagram shows that classes we will implement in our prototype. Under the name of each class are the associated parameters and functions. The relationship between the classes mainly involve directed association and composition. Also included in the UML class diagram is the multiplicity which shows whether the class has a zero-to-one, one-to-one or a one-to-many relationship. The UML class diagram includes create account, user login, enable and view alarm, add and view approved entrant, and send text messages to user.

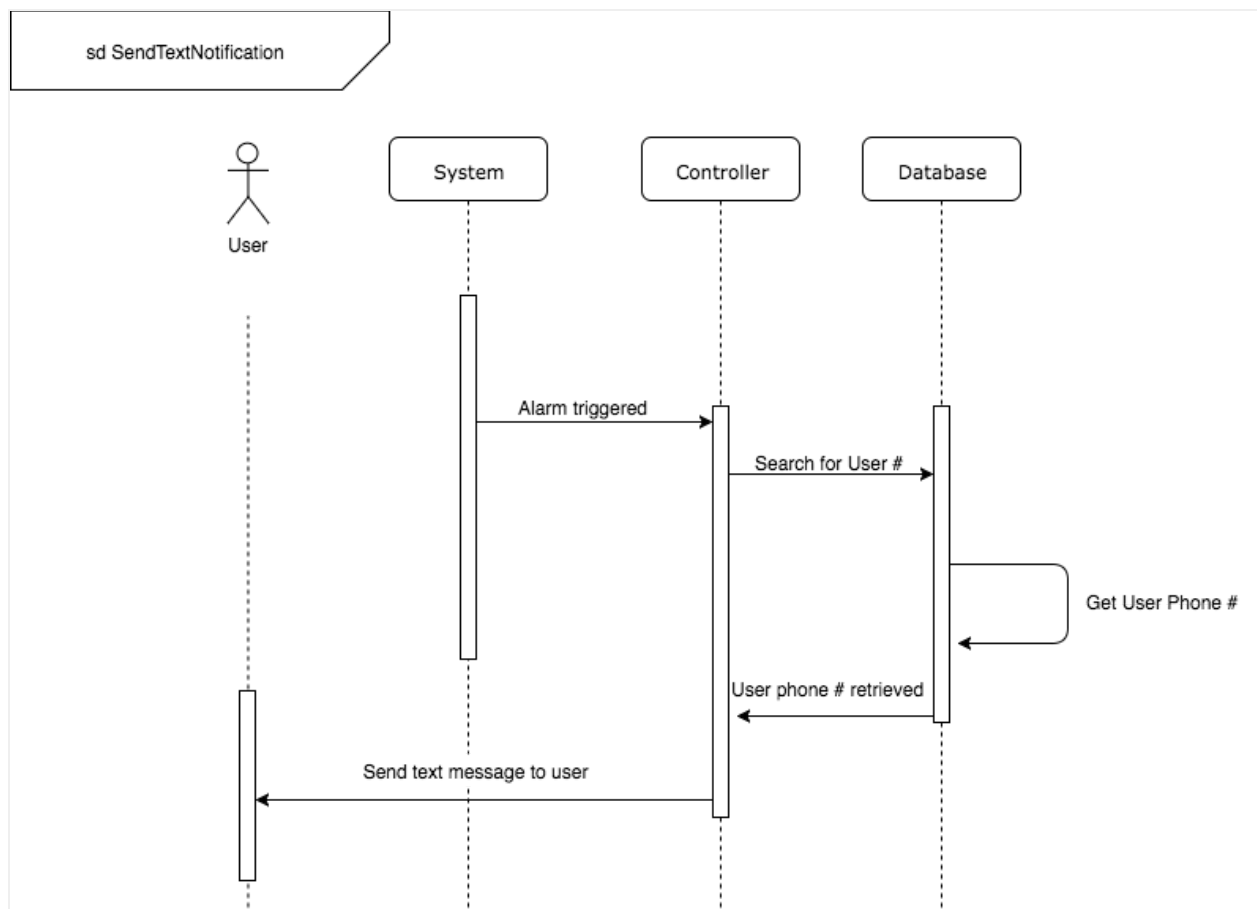
## Section 4: Sequence Diagram

### Login



When user opens Smart Security, they will land on the home screen. To use Smart Security, user must have a Google account to login successfully. User will enter their email address and password associated with their Google account and the backend system will check the database to ensure that the user input exists and if the credential matches. If any of the verification fails (credentials user enter is not in the database or password entered is wrong), the controller will send an error message back to the user screen. If credential is valid, user will be able to go onto the next UI screen, the dashboard.

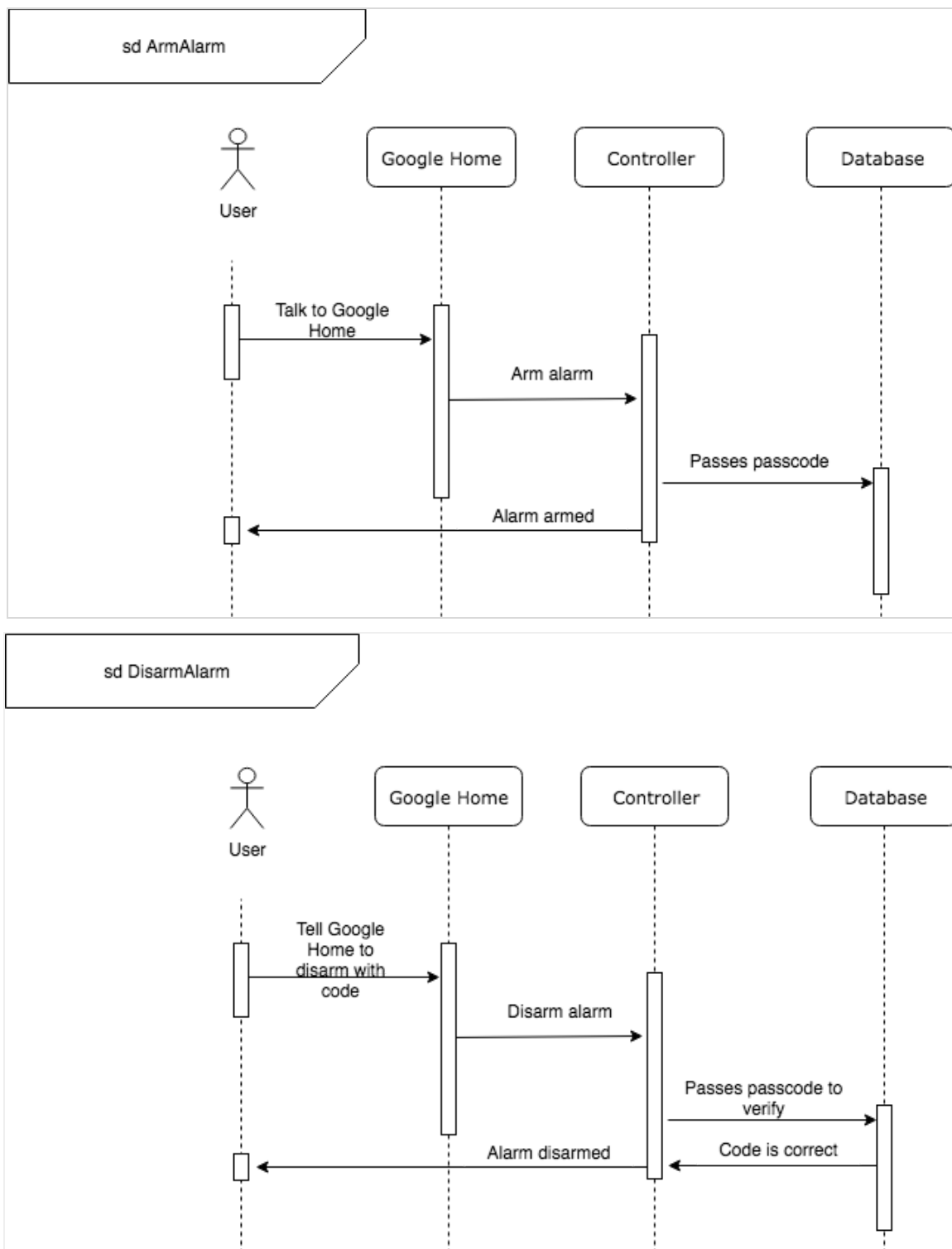
## Send Text Notification



When an alarm is tripped, the backend will search for the user's phone number in the database. The user's phone number is then retrieved and an alert text message will be sent to the user.



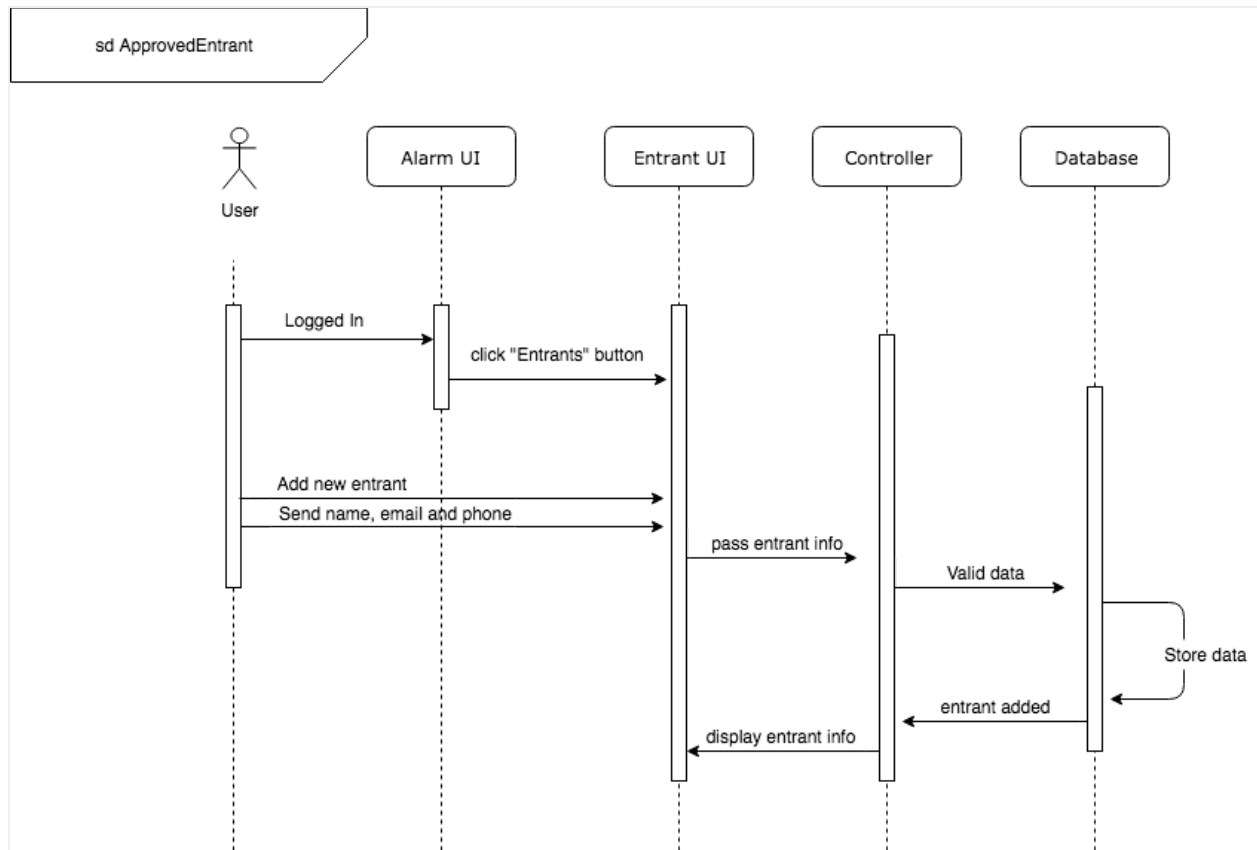
## Arm/Disarm Alarm



If the user wants to arm/disarm an alarm, the user will have to interact with the Google Home. They will have to speak to the Google Home, telling it to arm or disarm. In addition to telling Google Home what

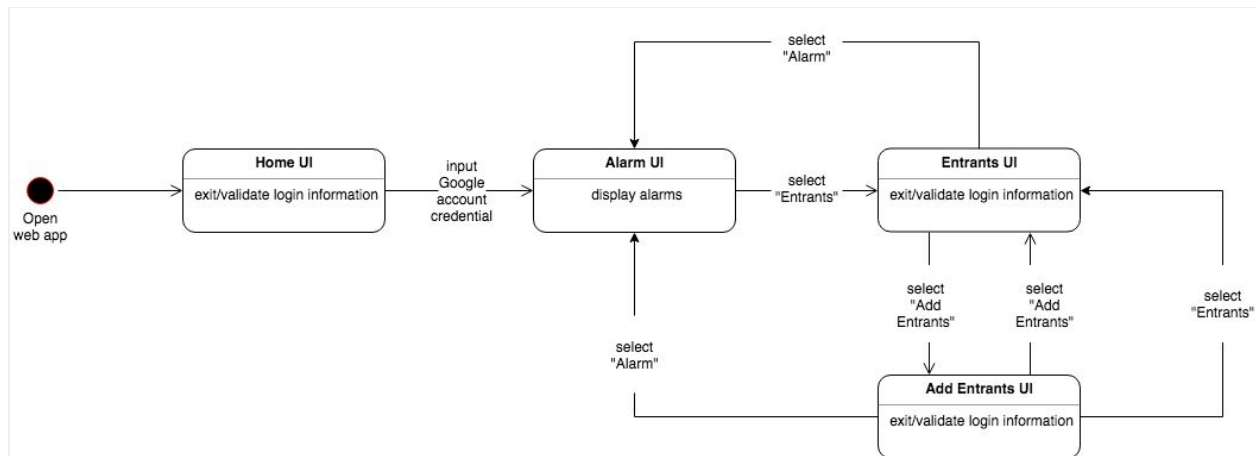
action to do, the user will have to provide a code. When arming the alarm, the code will be saved to the database. Once the user want to disarm the alarm, the code provided will be verified before the system disarms the alarm.

## Approve Entrant



If the user wants to view and/or add an entrant, the user will have to go to the approved entrant user interface. Once they are in the approved user interface, there will be a list of approved entrants. To add an entrant, the user will have to enter the entrant's name and phone number. The backend system will pass the data and the database will store that information. The entrant is then added and will be shown in the list of approved entrants.

## Section 5: State Machine Diagram



**Figure 1.** A state machine diagram for illustrating the different states of using the web application at each interface.

### Home UI

In this state, the user sees the Home Screen UI, which tells users about the application. There is a link on the right corner for users to log in with their Google account. Once the user is logged on with a valid account, the state changes to the “Alarm UI” state.

### Alarm UI

In this state, the user sees a log of past alarms, including the time they were set, the time they were tripped, and whether the entrant was authorized or unauthorized.

The user will be able to see the following buttons on the page:

- “Entrants” button - on selecting the “Entrants” button, the state transitions to the “Entrants UI” state

### Entrants UI

In this state, the user sees a list of their approved entrants.

The user will be able to see the following buttons on the page:

- “Alarm” button - on selecting the “Alarm” button, the state transitions to the “Alarm UI” state
- “Add Entrants” button - on selecting the “Add Entrants” button, the state transitions to the “Add Entrants UI” state

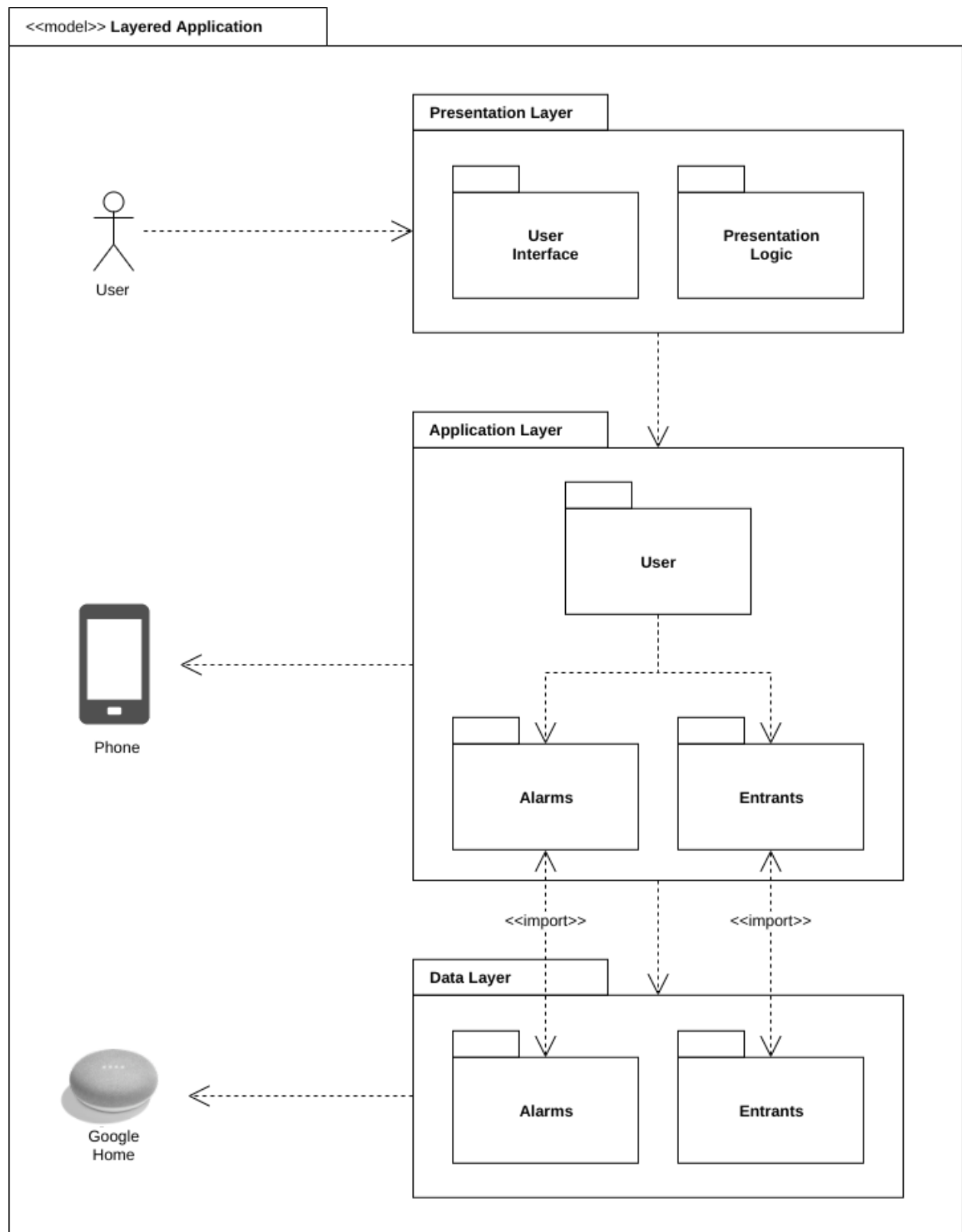
### Add Entrants UI

In this state, the user sees fields in which they can enter a new entrant’s information.

The user will be able to see the following buttons on the page:

- “Alarm” button - on selecting the “Alarm” button, the state transitions to the “Alarm UI” state
- “Entrants” button - on selecting the “Entrants” button, the state transitions to the “Entrants UI” state
- “Add Entrants” button - on selecting the “Add Entrants” button, the state transitions to the “Add Entrants UI” state

## Section 6: Package Diagram



## Presentation Layer

The user interface will be how the user interacts with the web application. The features that will be deployed in two weeks will be a part of the user interface. This includes interfaces for creating an account, logging in, enabling and viewing alarms, and adding and viewing entrants.

The presentation logic will handle the user interface functionality. This will also be responsible for sending a text to the user when an alarm is activated.

## Application Layer

The user class will contain their name, login credentials, email address, and phone number. Each user will also be associated with a list of alarms and entrants and the Google Home itself. The user class will most likely have to be associated with a Google account in order to connect to the Google Home.

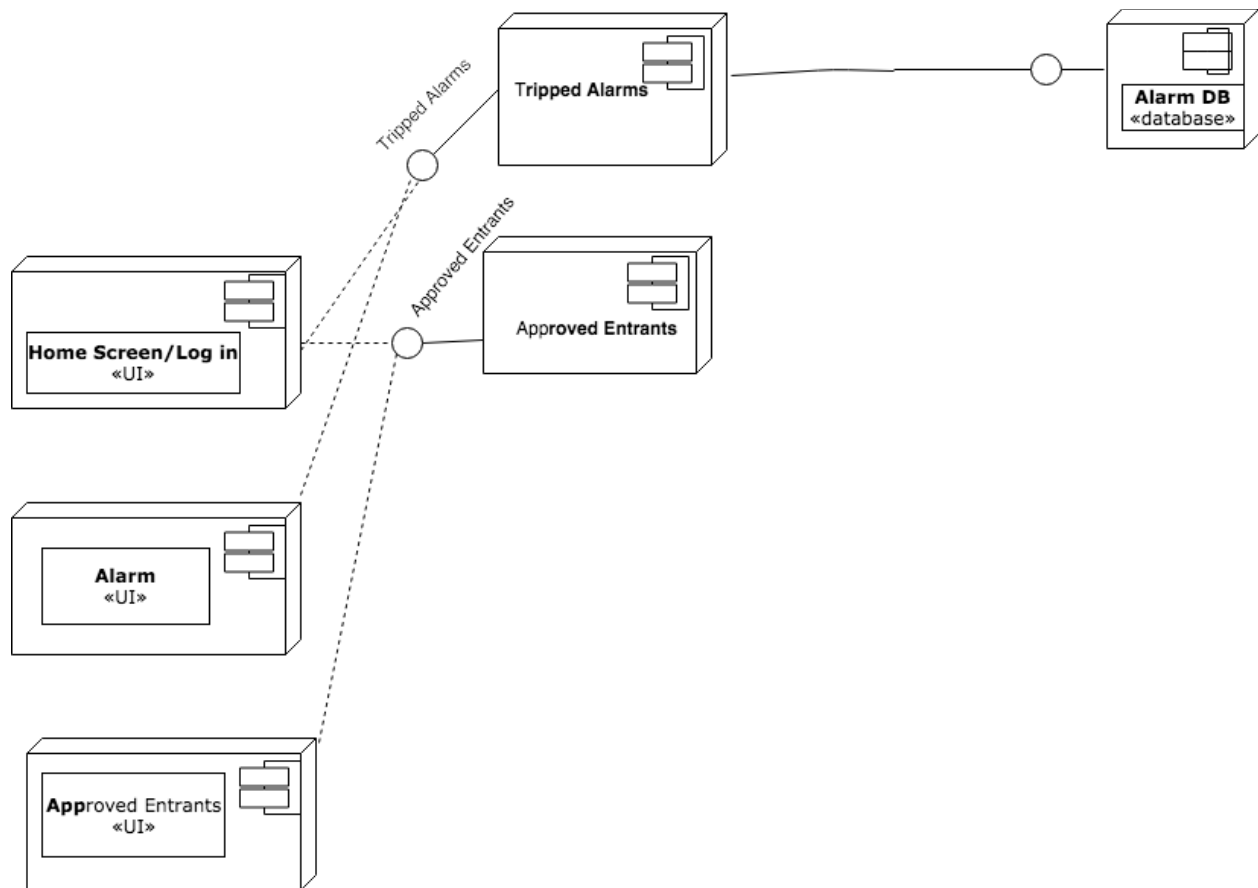
The alarms class will contain functionality for viewing alarms. It will also store a list of when the Google Home was activated. This information will be stored in a database.

The entrants class will contain functionality for adding and viewing entrants. Each entrant will contain their name, email, and phone number. This information will be stored in a database.

## Data Layer

The data for alarms and entrants will be stored in a database that will be accessed by the user through the web application and Google Home. The alarm database will contain a list of tripped alarms. The entrant database will contain a name, email, and phone number.

## Section 7: Component Diagram



## Home Screen UI

The Home Screen User Interface consists of the Account component which includes signing in using Google email address and acts as a dashboard for all other pages.

## Alarm UI

The Alarm UI is lead to by choosing the Tripped Alarms component from the Home Screen UI.

## Approved Entrants UI

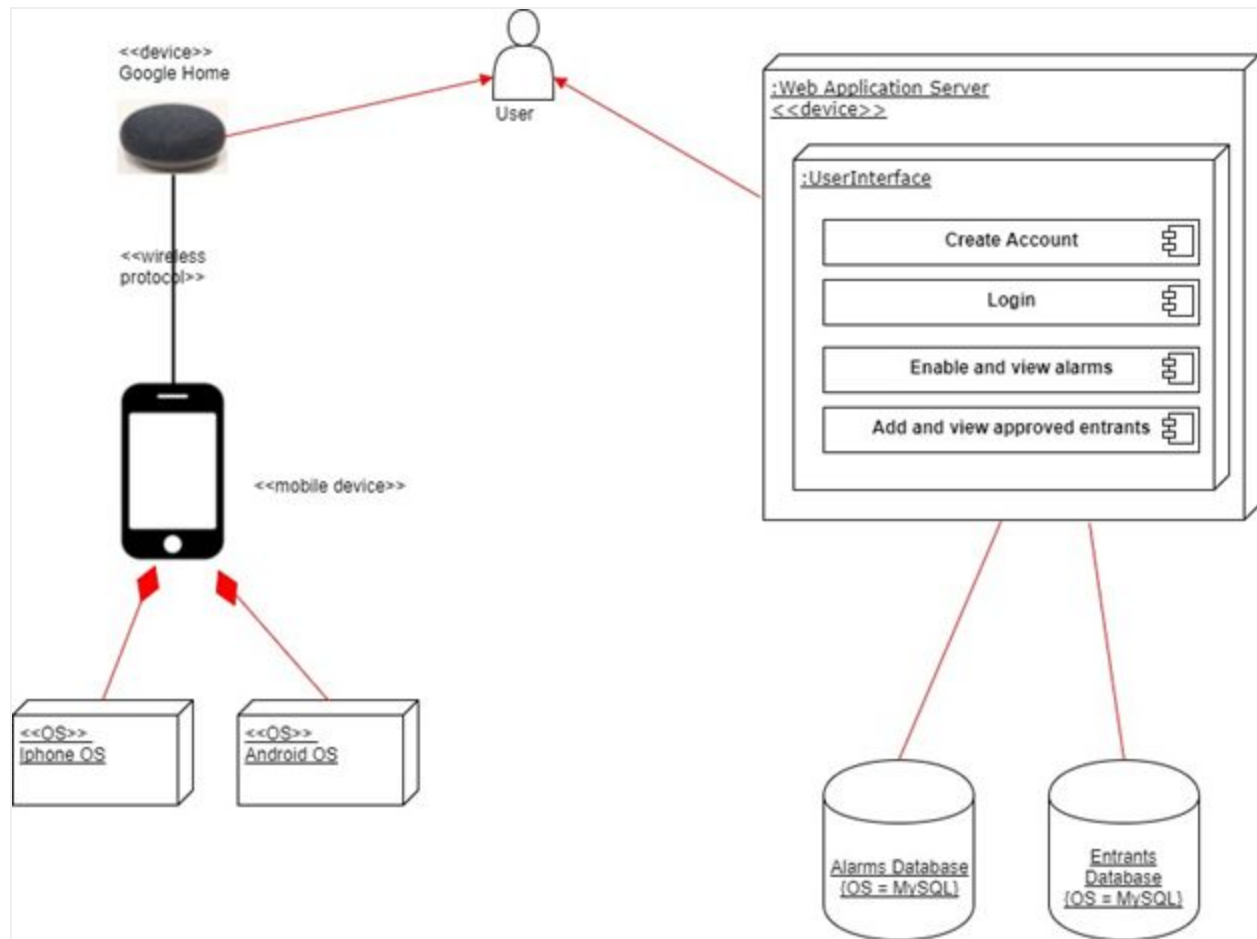
The Approved Entrants UI consists of choosing the Approved Entrants option from the Home Screen UI which will allow users to add, delete, and edit the entrants they would want approved.

## Alarm Database

The Alarm Database consists of the data gathered from the Account, Tripped Alarms, and Approved Entrants components.



## Section 8: Deployment Diagram



In this deployment diagram, we describe the hardware components that will be used in our project. For this project, the hardware component (represented by a node) will be the Web Application Server. The software components that will run on the Web Application Server are all of the user interface features. The user interface features that will be implemented on the Web Application Server are login, view alarms, and add and view approved entrants. The user will be able to use the different user interfaces to interact with the Web Application Server.

There will be two different databases, one for alarms and one for entrants. All the information that is stored in these databases will be accessed through the Web Application Server, and the user is able to get it.

In our project, we are using Google Home, which is a device that the user can communicate with and get information from. We are also using a mobile device that will use a wireless protocol to send/get information from Google Home. The mobile device (cell phone) will be either an iPhone or an Android depending on the user's phone.

## **Section 9: Implementation Plan**

For our implementation plan moving forward, Tiffany is working on the front-end of the web application adding and revising features to the interfaces for logging in, enabling/disabling the alarm, viewing the alarm history and adding/viewing approved entrants, as well as adding the sign in feature for logging in using a gmail address and password.. Helen is working on the backend side focusing on connecting the Google Home to the database. Connie, Dericka, and Josie are working on integrating Google Home using API and voice recognition. Lastly, Aruna is working on documenting of the web application.

## **Conclusion**

In conclusion, we were able to revise more detailed designs and models for our project. For this assignment previously, we developed a UML class diagram, a sequence diagram, a state machine diagram, a package diagram, a component diagram, and a deployment diagram, and are continuing to revise these diagrams to fit the states of our project. We also decided on our current implementation platform and techniques, created a better implementation plan for the features defined in this assignment for the upcoming weeks. With this assignment, we gathered more details to help with our implementation.