

# **Hausmate Software Requirements Specifications**

Adam Hirata, Jared Rebuyon, Miliano Mikol, Raymond Law

## **Table of Contents**

- 5.0 Requirements Specification
  - 5.1 Introduction
  - 5.2 CSCI Component Breakdown
    - 5.2.1 GUI Components
    - 5.2.2 Database Components
    - 5.2.3 Server Components
  - 5.3 Functional Requirements
  - 5.4 Performance Requirements
  - 5.5 Mock Up Based on CSC Specification
  - 5.6 Project Environment Requirements
    - 5.6.1 Project Development
    - 5.6.2 Project Execution

## **5.0 Requirements Specification**

### **5.1 Introduction**

One-in-three U.S. adults have an adult roommate. This number is growing. Among the many causes for roommate conflicts, miscommunication and even no communication are especially problematic. This often leads to gossip, passive aggression, and a tense living environment. To promote friendships and proper communication, we introduce the concept of Hausmate. Computers offer unique capabilities for making communication easier. While it is true that confronting problems in person is beneficial, this is, in many cases, ideal as it does not factor potential schedule conflicts. Regardless, this application is no substitute for in-person interactions. It is a platform for general organization and communication that aids relational management.

This document outlines the software requirements specifications for Hausmate.

## 5.2 CSCI Component Breakdown

Hausmate is composed of the following CSCs:

<b>5.2.1</b>	<b><u>GUI CSC</u></b>	-	Mobile/Desktop application to be interacted upon by the user
5.2.1.1	Login Screen	-	Screen where the user is able to Create or join a Haus
5.2.1.1.1	Haus Creation Module	-	Button that directs the user to create a Haus
5.2.1.1.2	Haus Joining Module	-	Button that directs the user to join a Haus
5.2.1.2	Main Screen	-	Screen where the user is able to see all offered tools and preview modules
5.2.1.2.1	Settings Menu	-	Menu where user is able to change app and profile settings. Has a main menu module which is a clickable button
5.2.1.2.1.1	Setting Module	-	Each individual setting will have its own interactive module (e.g. toggles, sliders, input, etc.)
5.2.1.2.2	Profile Menu	-	Menu where user can edit their

profile information and status

- 5.2.1.2.3 Events Module - Display preview of upcoming events on the Main Screen
- 5.2.1.2.4 Lists Module - Displays pinned lists on the Main Screen
- 5.2.1.2.5 Announcements Module - Displays recent and pinned Announcements on the Main Screen
- 5.2.1.2.6 Chat Module - Displays recent and unread Messages on the Main Screen
- 5.2.1.2.7 Status Module - Displays statuses and personal messages of all Hausmates on the Main Screen
- 5.2.1.2.8 Tools - Pull-out list from the main menu which lists button modules for each individual tool
- 5.2.1.2.8.1 Calendar Tool - Tool where the user is able to edit a calendar that can be seen by their housemates. Has a main menu module which displays the day/day of week and upcoming events

- 5.2.1.2.3.1.1 Create/Edit event event - Screen to create a new event which includes, but is not limited to, setting an event name, date and time, recurrence rules, etc. with corresponding sub-modules for altering these properties
- 5.2.1.2.8.2 List Tool - Tool where the user is able to edit and create Lists that can be seen by their housemates. Has a main menu module which displays "pinned" lists
- 5.2.1.2.8.2.1 Individual List(s) - all From the list menu, displays lists and a preview of each list's first few components. Clicking will lead to a new screen which displays the list in full
- 5.2.1.2.8.2.2 List Component - A text representation of a list item. Will comprise sub-modules for deleting this item from a list
- 5.2.1.2.8.2.3 Quick-Add Item of - An input module at the bottom of each list which allows user to quickly append a new item to a list

- 5.2.1.2.8.2.4 Create/Edit List - Screen to create/edit new lists and corresponding list items. Will have sub-modules to alter properties of items. Will have a button module from the List Menu
- 5.2.1.2.8.3 Chat CSU - Tool where the user is able to directly send and receive text messages from their housemates via text input. Has a main menu module which displays the last received message or the number of unopened notifications
- 5.2.1.2.8.3.1 General Chat CSU - Displays all group chat messages and comprises a sub-module to write and send a new message
- 5.2.1.2.8.3.2 Direct Message CSU - Displays all private messages between two users and comprises a sub-module to write and send a new message
- 5.2.1.2.8.3.3 Members List Module - List of all members in the house as sub-module buttons. Clicking will entail the Direct Message CSU
- 5.2.1.2.8.4 Announcements Tool - Tool where the user can send an

announcement message that can be seen by all housemates. Used for messages with more importance. List all announcements in order of chronology with the option of pinning a single announcement. Has a main menu module which lists the most recent announcements

#### 5.2.1.2.8.4.1 Announcement Addition

- Text input module which allows users to type an announcement and send it to the whole house

#### 5.2.1.2.8.5 Anonymous Board Tool

- Messaging system that allows user to send messages to others without revealing the user's name. Lists messages in order of chronology

#### 5.2.1.2.8.5.1 Anonymous Message Addition

- Text input module which allows users to type a message and send it to the whole house anonymously



## **5.2.2 Database CSC**

### 5.2.2.1 Haus Entity

#### 5.2.2.1.1 id

#### 5.2.2.1.2 name

#### 5.2.2.1.3 calendar

### 5.2.2.2 User Entity

#### 5.2.2.2.1 id

#### 5.2.2.2.2 first\_name

#### 5.2.2.2.3 last\_name

#### 5.2.2.2.4 status

### 5.2.2.3 List Entity

#### 5.2.2.3.1 id

#### 5.2.2.3.2 type

#### 5.2.2.3.3 name

### 5.2.2.4 List Entry Entity

#### 5.2.2.4.1 id

#### 5.2.2.4.2 title

- A centralized storage system for user data
- An entity that groups many users.
- Unique ID of a Haus that acts as a primary key (type: int)
- Name of Haus (type: varchar)
- Stores a shared calendar reference key based on the Google Calendar API (type: varchar)
- An entity that stores user profile information
- Unique ID of user that acts as a primary key (type: int)
- The first name of a user (type: varchar)
- The last name of a user (type: varchar)
- Current toggle status of user (type: int)
- An entity that contains many List Entries
- Unique ID of a list that acts as a primary key (type: int)
- A category for the list (type: int)
- Name of list (type: varchar)
- An entity describing a single entry in a list
- Unique ID of a list entry that acts as a primary key (type: int)
- An entry's title, subject, or description (type: varchar)

- 5.2.2.4.3 checked
  - An optional boolean that indicates a completed list entry (type:boolean)
- 5.2.2.5 Users Table
  - A defined relation that relates one Haus to many Users
- 5.2.2.5.1 haus\_id
  - The primary key that relates a Haus key to a user foreign key (type: int)
- 5.2.2.5.2 user\_id
  - The foreign key of a user (type: int)
- 5.2.2.6 Lists Table
  - A defined relation that relates one Haus to many Lists
- 5.2.2.6.1 haus\_id
  - The primary key that relates a Haus key to a list foreign key (type: int)
- 5.2.2.6.2 list\_id
  - The foreign key of a list (type: int)
- 5.2.2.6 Entry Table
  - A defined relation that relates many Lists to many List Entries
- 5.2.2.6.1 list\_id
  - The primary key that relates a List key to a list entry foreign key (type: int)
- 5.2.2.6.2 entry\_id
  - The foreign key of a list entry (type: int)

### **5.2.3**      **Server CSC**

- 5.2.3.1      Authentication      -      An system that verifies user credentials based on a supplied email/password or third-party verification
- 5.2.3.2      Notification System      -      A simple notification system that can track and send push notifications to users when an event occurs or a message is sent
- 5.2.3.4      Database Server      -      A cloud-hosted database for storing or querying information
- 5.2.3.5      Database System      -      A system for establishing a schema and client-server interactions between the Hausmate application and the Database server
- 5.2.3.6      Google Calendar API      -      Asynchronous interactions that store and retrieve data through the Google Calendar API

### **5.3 Functional Requirements by CSC**

#### 5.3.1 Graphical User Interface Functional Requirements

##### 5.3.1.1 The Login Screen shall display two Buttons

The first will be known as the Create a Haus Button

The second will be known as the Join a Haus Button

##### 5.3.1.2 The Create a Haus button shall prompt and guide the user on

creating a Haus

##### 5.3.1.3 The Join a Haus button shall prompt and guide the user on joining a Haus

##### 5.3.1.4 The GUI subsystem shall display a screen for the main Application.

This will be known as the 'Main Screen'

##### 5.3.1.5 The Main Screen shall display information of Hausmate Statuses

This will be known as the Status Module

##### 5.3.1.6 The Main Screen shall display recent announcements

This will be known as the Announcements Module

##### 5.3.1.7 The Main Screen shall display Calendar and upcoming events

This will be known as the Events Module

##### 5.3.1.8 The Main Screen shall display uncompleted Lists

This will be known as the Lists Module

##### 5.3.1.9 The Main Screen shall display unread direct messages

This will be known as the Chat Module

##### 5.3.1.10 The Main Screen shall contain a button for the Settings Menu

This will be known as the Settings Button

##### 5.3.1.11 The Main Screen shall contain a button for editing profiles

This will be known as the Profile Button

##### 5.3.1.12 The Main Screen shall contain a pull out menu containing all tools

This will be known as the Tool Pull Out Menu

##### 5.3.1.13 The Profile Button shall transition to a new screen

This screen will be known as the Profile Screen

- 5.3.1.14 The user shall be able to change their profile picture in the Profile Screen
- 5.3.1.15 The user shall be able to change their status in the Profile Screen
- 5.3.1.16 The user shall be able to change their personal message in the Profile Screen
- 5.3.1.17 The user shall be able to change their displayed name in the Profile Screen
- 5.3.1.18 The Settings Menu button shall transition to a new screen This will be known as the Settings Screen
- 5.3.1.19 The user shall be able to change notification settings in the Settings Screen
- 5.3.1.20 The user shall be able to change the display theme in the Settings Screen
- 5.3.1.21 The user shall be able to change displayed Modules of the Main Screen in the Settings Screen
- 5.3.1.22 The user shall be able to Logout in the Settings Screen
- 5.3.1.23 The user shall be able to leave their current Haus in the Settings Screen
- 5.3.1.24 The Tool Pull Out Menu shall display a tool to edit the calendar  
This will be known as the Calendar Tool
- 5.3.1.25 The Tool Pull Out Menu shall display a tool to edit the lists  
This will be known as the Lists Tool
- 5.3.1.26 The Tool Pull Out Menu shall display a tool to make an announcement  
This will be known as the Announcements Tool
- 5.3.1.27 The Tool Pull Out Menu shall display a tool to send and view direct messages  
This will be known as Chat Tool
- 5.3.1.28 The Tool Pull Out Menu shall display a tool to send and view anonymous messages

- This will be known as the Anonymous Board Tool
- 5.3.1.29 The Calendar Tool shall display all upcoming events by date and time
- 5.3.1.30 The Calendar Tool shall contain a button to create a new event
- 5.3.1.31 The Calendar Tool shall allow the user to edit created events
- 5.3.1.32 The Calendar Tool shall allow the user to remove created events
- 5.3.1.33 Adding, editing, and removing events in the Calendar Tool shall update the Events Module on the Main Screen
- 5.3.1.34 Any changes made using the Calendar Tool will sync with the Calendar Tool of all other Hausmates
- 5.3.1.34 The Lists Tool shall display all created Lists
- 5.3.1.35 Interacting with a created list shall allow the user to check off list items
- 5.3.1.36 The Lists Tool shall contain a button to create a new list There will be 2 List Types, Recurring and Non-Recurring
- 5.3.1.37 A Recurring List shall refresh after a specified time frame
- 5.3.1.38 A Non-Recurring List shall stay checked off until changed by a user
- 5.3.1.39 The Lists Tool shall allow the user to edit created lists
- 5.3.1.40 The Lists tool shall allow the user to delete created lists
- 5.3.1.41 Changes made in the Lists Tool shall update the Lists Module on the Main Screen
- 5.3.1.42 Any changes made using the Lists Tool will sync with the Lists Tool of all other Hausmates
- 5.3.1.43 The Announcements Tool shall display past announcements
- 5.3.1.44 The Announcements Tool shall display which user sent an announcement
- 5.3.1.45 The Announcements Tool shall allow the user to send an

announcement

- 5.3.1.46 The Announcements Tool shall ask the user to confirm sending announcement
- 5.3.1.47 The Announcements Tool shall allow the user to "pin" one single announcement
- 5.3.1.48 The Announcements Tool shall allow the user to "unpin" a pinned announcement
- 5.3.1.49 Changes made in the Announcements Tool shall update the Announcements Module on the Main Screen
- 5.3.1.50 Any changes made using the Announcements Tool will sync with the Announcements Tool of all other Hausmates
- 5.3.1.51 The Chat Tool shall display past messages of the house
- 5.3.1.52 The Chat Tool shall display which user sent a message (non-direct message)
- 5.3.1.53 The Chat Tool shall allow the user to send a message to all members of a house at once
- 5.3.1.54 The Chat Tool shall allow the user to send a message to another house member
- 5.3.1.55 Changes made in the Chat Tool shall update the Chat Module on the Main Screen
- 5.3.1.56 Any changes made using the Chat Tool shall sync with the Chat Tool of applicable Hausmates
- 5.3.1.57 The Chat Tool shall send push notifications to applicable Hausmates
- 5.3.1.58 The Anonymous Board Tool shall display past anonymous messages
- 5.3.1.59 The Anonymous Board Tool shall allow the user to send an anonymous message to one Hausmates
- 5.3.1.60 The Anonymous Board Tool shall allow the user to send an anonymous message to all Hausmates
- 5.3.1.61 The Anonymous Board Tool shall allow the user to delete an anonymous message
- 5.3.1.62 Any changes made using the Anonymous Board Tool shall sync with the Anonymous Board Tool of applicable Hausmates

5.3.1.63 The Anonymous Board Tool shall send a push notification to applicable Hausmates

## 5.3.2 Database Functional Requirements

5.3.2.1 A new user entity shall be created with a unique identifier upon account creation

The unique identifier will be either the user's email or a third party service of their choice

5.3.2.2 Hauses shall be queryable if the user opts to join an existing Haus

5.3.2.3 A new Haus entity shall be created if the user opts to not join an existing Haus

5.3.2.4 A new List entity shall be created for every list the user creates and saves

5.3.2.5 A new List Entry entity shall be created for every entry the user adds to a list

The list entry will be associated with the respective list to which the user is adding

5.3.2.6 Each Haus shall store one reference to a Google calendar  
The Google calendar will be viewable and editable by all users in the Haus

5.3.2.7 The database shall delete the respective User entity of the user who opts to delete their account

5.3.2.8 The database shall delete the respective Haus entity of the users who completely abandon their Haus

Abandonment will be determined by whether the users opt to delete the Haus

Abandonment will also be determined by whether all users have left the Haus



- 5.3.2.9 The database shall delete the respective List entity of the List that a user opts to delete
- 5.3.2.10 The database shall delete the respective List Entry entity of the entry that a user opts to delete
- 5.3.3 Server Functional Requirements
  - 5.3.3.1 The server shall authenticate user login credentials
  - 5.3.3.2 The server shall secure and protect user login credentials
  - 5.3.3.3 The server shall track login history
  - 5.3.3.4 The server shall send automatic push notifications to user's devices
  - 5.3.3.5 The server shall record sent notifications
  - 5.3.3.6 The server shall listen for events that trigger notifications
  - 5.3.3.7 The server shall query a database server for stored information
  - 5.3.3.8 The server shall add entries to a database server to store information
  - 5.3.3.9 The server shall interact with the phone client to send and receive information through endpoints
  - 5.3.3.10 The server shall validate all received information
  - 5.3.3.11 The server shall send appropriate error message when invalid information is received
  - 5.3.3.12 The server shall interact with Google Calendar API
  - 5.3.3.13 The server shall update shared Lists, Calendars, Statuses, and Announcements of all Hausmates if a change is made

- 5.4 Performance Requirements by CSC
- 5.4.1 It will take no longer than 5 seconds for the application to startup
- 5.4.2 It will take no longer than 5 seconds to authenticate a user
- 5.4.3 It will take no longer than 5 seconds to create and store a new account
- 5.4.4 It will take no longer than 5 seconds to query the database for a Haus
- 5.4.5 It will take no longer than 3 seconds to log out
- 5.4.6 It will take no longer than 3 seconds to join a Haus
- 5.4.7 It will take no longer than 10 seconds to send a message
- 5.4.8 It will take no longer than 5 seconds to refresh a page
- 5.4.9 It will take no longer than 1 second to transition between pages
- 5.4.10 It will take no longer than 0.1 seconds for an interactive GUI element to react to user interaction where no feedback is necessary except to display a result
- 5.4.11 It will take no longer than 10 seconds for shared calendars to sync across all devices
- 5.4.12 It will take no longer than 10 seconds for shared lists to sync across all devices
- 5.4.13 It will take no longer than 10 seconds for shared messages to sync across all devices
- 5.4.14 It will take no longer than 10 seconds for user statuses to sync across all devices
- 5.4.15 It will take no longer than 3 seconds to create a calendar

event

- 5.4.16 It will take no longer than 3 seconds to create a shared list
- 5.4.17 It will take no longer than 10 seconds for an announcement to be sent
- 5.4.18 It will take no longer than 10 seconds for a push notification to be sent to a user's device
- 5.4.19 It will take no longer than 10 seconds to resolve an API request
- 5.4.20 The application will support 50 concurrent users at a minimum
- 5.4.21 The application will store messages no older than one month

[illegible]

## 5.6 Project Environment Requirements

### 5.6.1 Development Environment Requirements

The following are the hardware requirements to develop Hausmate:

Category	Requirement
Processor	Dual-Core Intel Core i5
Hard Drive Space	10MB
RAM	128MB
Display	800x600, 256 colors
Sound Card	Optional

These requirements are based off of the oldest computer model utilized in the development team. The computer is recent enough to support the minimum software requirements for development tools and packages. Any model older or less-capable than these specs may not support the development tools used for Hausmate.

The following are the software requirements to develop Hausmate:

Category	Requirement
Operating System	Windows 10, MacOS Catalina
Compiler/Language	React Native
Graphics	Adobe Photoshop

Windows 10 and MacOS Catalina are required for this project. This is because most development tools require these software versions at a minimum in order to operate. There will be no attempt to test the program under Windows 10 or MacOS Catalina.

The program will be written in React Native using Microsoft Visual Studio, Expo CLI, and AWS Amplify CLI.

Adobe Photoshop will be used to produce the graphical images for the project.

### 5.6.2 Execution Environment Requirements

The following are the hardware requirements to execute Hausmate:

Category	Requirement
Phone	iPhone X
Memory	1GB
RAM	3GB
Chipset	Apple A11 Bionic (10nm)
CPU	Hexa-core 2.39GHz (2x Monsoon + 4x Mistral)
GPU	Apple GPU (three-core graphics)

The iPhone X contains all minimum functionalities and software required to support the widget-heavy features of Hausmate.

The following are the software requirements for Hausmate:

Category	Requirement
-----	-----
iOS	14.0.1
Compiler/Language	Javascript

An iPhone X with iOS 14.0.1 is required for this project as it contains security patches along with support for the programming languages that will be used. Because of the scope of the project, there will be no attempt to test the program on hardware below an iPhone X under version 14.0.1