XXXX  
---Hobby Matcher  
  
Hobby Matcher team  
**Authors:** Chris Boffa, Haoxuan Li, Xueshi Wang, Zhe Sun (in alphabetical order)

|  |  |  |
| --- | --- | --- |
| Version | Date | Description |
| 1.0 | Oct 12, 2018 | 1. Mission statement: determine the major goal of this apartment finder, who are our customer, what can it function, how it function. Who do we stand for? Why do we do this? 2. Flow chart: confirm the working flow process, from brief register, to whether find apartment first or group roommate first. |
| 1.1 | Oct 21, 2018 | 1. Found out all the stakeholders concerning of our application.  2. Acquire some requirements from some of the stakeholders by asking questions from them.  3. Created the first edition of the requirement specification. |
| 1.2 | Oct 30, 2018 | 1. Created 6 user cases, and discusses about the deeper logical processes of the whole working flow. |
| 1.3 | Nov 17, 2018 | 1. Recreated the user cases and added one, and redefine some working processes, reach an agreement on the application's name, working flows.  2. Acquired all the non functional requirements 3. Drafted a draft about purpose, project scope, and product perspectives. 4. Recreated the flow chart. |
| 1.4 | Nov 24, 2018 | 1. Reach an agreement on application interface. |
| 1.5 | Nov 29, 2018 | 1. Recreated the flow chart.  2. Prepared for the presentation, made slides. |
| 1.6 | Nov 30, 2018 | 1. Completed the prototypes.  2. Modify the details of all parts.  3. Corrected the grammar aspect of the requirement specification. And review the final version document. |

**Contents**

[1. Mission Statement](#_gjdgxs)

[1.1 Background](#_30j0zll)

[1.2 Objectives](#_1fob9te)

[1.3 Project scope](#_3znysh7)

[1.4 References](#_2et92p0)

[2. Overall Description](#_tyjcwt)

[2.1 Product perspective](#_3dy6vkm)

[2.2 Operating environment](#_1t3h5sf)

[2.3 Assumptions and dependencies](#_4d34og8)

[2.4 Modes of Operation](#_2s8eyo1)

[3. Key Stakeholders](#_17dp8vu)

[3.1 College students](#_3rdcrjn)

[3.2 College staffs](#_26in1rg)

[3.4 Project manager](#_lnxbz9)

[3.5 Developers](#_35nkun2)

[3.6 Advertisers](#_1ksv4uv)

[3.7 Housing agencies](#_44sinio)

[4. Key Constraints](#_2jxsxqh)

[4.1 System constraints](#_z337ya)

[4.2 Stakeholders constraints](#_3j2qqm3)

[4.3 Others](#_1y810tw)

[5. Key Requirements](#_4i7ojhp)

[5.1 Ways of eliciting requirements](#_2xcytpi)

[5.2 User Requirements](#_1ci93xb)

[5.2.1 Customers](#_3whwml4)

[5.2.2 Project manager](#_2bn6wsx)

[5.2.3 Developers](#_qsh70q)

[5.2.4 Sponsors](#_3as4poj)

[5.2.5 Advertisers](#_1pxezwc)

[5.3 Business Requirements](#_49x2ik5)

[5.3.1 Competitors](#_2p2csry)

[5.3.2 Business Goal](#_147n2zr)

[5.3.3 Other requirements](#_3o7alnk)

[5.4 Others](#_23ckvvd)

[6. Key Use Cases](#_ihv636)

[6.1 Registration](#_32hioqz)

[6.2 User log in](#_1hmsyys)

[6.3 Publish Apartment](#_41mghml)

[6.4 Input personal information](#_2grqrue)

[6.5 Search apartment](#_vx1227)

[6.6 Find roommates](#_3fwokq0)

[6.7 My profile](#_1v1yuxt)

[7. Quality Attributes](#_4f1mdlm)

[7.1 Usability](#_2u6wntf)

[7.2 Performance](#_19c6y18)

[7.3 Security](#_3tbugp1)

[7.4 Reliability](#_28h4qwu)

[8. Flow Charts](#_nmf14n)

[8.1 Overall](#_37m2jsg)

[8.2 Register](#_1mrcu09)

[8.3 Login](#_46r0co2)

[8.4 Publish Apartment](#_2lwamvv)

[8.5 Search Apartment](#_111kx3o)

[9. UI Prototypes](#_3l18frh)

[9.1 Homepage](#_206ipza)

[9.2 Login](#_4k668n3)

[9.3 Input Personal Information](#_2zbgiuw)

[9.4 Search Roommates](#_1egqt2p)

[9.5 Publish Detail Information](#_3ygebqi)

[9.6 Search Apartments](#_2dlolyb)

[9.7 Search Apartments Result](#_sqyw64)

[9.8 Favorite Apartment](#_3cqmetx)

[9.9 My Roommates](#_1rvwp1q)

[9.10 Published Apartment](#_4bvk7pj)

[9.11 Message](#_2r0uhxc)

[9.12 My Information](#_1664s55)

[10. Questionnaires](#_25b2l0r)

[11. Role Play](#_kgcv8k)

[11.1 Roll as Users to another team’s project](#_34g0dwd)

1. Mission Statement

1.1 Background

Most people have their own hobbies, but when there is a situation, the hobby which you are very into, not the thing to the friends around you, they may be addicted to basketball, or maybe tennis, but just not yours, and you want to play with people who have the same hobby like you, here comes the problem, how could you find those people? there is no effect way to solve that problem so far.

1.2 Objectives

To help people who have hobbies to find people who have the same hobbies like them, and are able to organize events or participate in events, to enjoy the fun of hobbies better.

1.3 Project scope

1: To provide the service for user to find people who have the same hobby as them

2: To provide the service for user to participate in events and hold events

3: To provide the service for user to share their personal experience or stories with other user

1.4 References

1. XXXXXXXXX

2. XXXXXXXXX

3. XXXXXXXXX

2. Overall Description

2.1 Product perspective

1.Users can find the people who has the same hobby as them, which could base on the current location, or any customized location. At the beginning, user have to choose or register their own hobbies, and enter some specific location information, such as city, zip code, etc. And then the system will bring the hobbies that you chose to the front of the list, you can browse them or other hobbies.

User will have an option to be a coach or not, if the user think they are professional, they can tag themselves a coach, and could earn money by being a coach, they could also set up a status, like how professional they are, from beginner to expert, which will show on their personal page.

2. Once the user chooses a particular hobby, the website will jump to a new page, which entails recent activities, activities near me, active users, active hobby groups, and popular tweets from the up to down.

3.When the user clicks on the activities, the page will show all the information to the user, such as activities pictures if any, activities introduction, previous live pictures if any, specific location and time, and event organizer, etc. You could choose to enroll in it, or to mark it.

Once you enrolled successfully, you could see the contact information of that event holder.

4. When the user checking the active users, they can click on their facial photo, and then a new page pops up, showing the public information shared by this user, including gender, personal introduction, and current general locations and activities they have joined before.

5, When the user chooses to have a look at the active groups which is like bbs, they could check the hobby groups’ introduction, and if interested, they can request to join that group/talk.

6, User could also read the tweets freely, over there, they can comment on the passage or others.

2.2 Operating environment

The website could be assessed on any computer web browser, and it UI should be also adjusted to the smartphone users, compatible with smart phone browser.

2.3 Assumptions and dependencies

1.The project members are familiar with Agile development and good at Java EE, HTML, CSS and JavaScript.  
2. The budget of app is 50 thousand dollars.  
3.The process time of requirement is three months.  
4.The project schedules should be accurate to the day.

2.4 Modes of Operation

1. Standard Mode: The website runs normally; all users can use it.   
2.Maintenance Mode: Website runs, but only the part being operated may be shutdown. Only technician can access to the system.  
3.Backup Mode: Web server runs normally, all users can use it, technician will access to the core server to make system backups manually.

3. Key Stakeholders

3.1 General User

This website is designed for anyone who has a hobby and offer them a chance to get together have activities, meetups, and sharing opinions like experience and stories.

3.2 Project manager

Project manager will control and lead the team to accomplishment the project. The manager wants the product to be delivered with all the primary requirements met. And hope the cost could be as lower as possible.

3.3 Developers

Agile development group will be responsible for developing this application. For developers, they want this application’s architecture to be clear enough and well structured. In addition, all features should be reasonable and practical, they don’t want to face the ambiguity software requirements. The system should be reliable and stable so that those maintenance personnel don’t have to spend much time on maintenance or refactoring.

3.4 Advertisers

In our project, the advertisers are one of the most important stakeholders. We get money from them to keep our website running, to rent server, and pay the clerks’ salary.

3.6 XXXX

XXXXXX.

3.7 XXXXXX

XXXXXXXX.

4. Key Constraints

4.1 System constraints

1.The website should be able to display properly on any smart phones devices’ web browser.

2.The website should be secure enough to protect the users’ personal information from hacking.

4.2 Stakeholders constraints

1. Users will be verified by their email addresses or cell phone number.  
2. Users need to follow the rules of the application.

4.3 Others

1.The app will meet the requirement of local law, government standards and policy.

5. Key Requirements

5.1 Ways of eliciting requirements

1. Passing out questionnaires

2. Interview on the spot

3. Telephone interview

4. Experience

5.2 User Requirements

5.2.1 Customers

1. Users could sign up with their email addresses or customized user names and their customized passwords. During the signup process, user have to set up three security questions to ensure user will be able to find their password back some day.

2, Users could log in with their registered email addresses or user names combined with their passwords.

3, If the user forgot their password, they could retrieve their passwords by receiving an email to their registered email address from our system. If user don’t have their email address, they have another option, which is to answer one of the three security questions, if the answer is correct, they are able to reset their password, or rather failed, but they can try this option as many times as they want.

2. Once user log into the website, for the first time, they have to edit their personal profile. Which includes:

Nickname(required), gender, date of birth, hobby fields, self introduction(required). These are listed from up to down.

Plus:

Nickname is not allowed to repeat, which could be any character, like chinese, english, spanish, etc.

Gender here is offered six options, including cisgender, genderqueer/gender non-conform, man, other, trans/transgender, woman.

Hobby fields this option will give user many choices, basketball, tennis, or something else, user could find their hobbies by searching, if the user doesn’t find their hobby out their, they are allowed to add one to the system, and tag it on themself. Which means all the hobbies the user had choosened will be shown on this user’s personal main page.

Self introduction is required for the user to fill, at least fifteens words.

The user is also given a choice to set the status of their profession level, whether a beginner or the professional guy.

3. Participate in activities. This means user can check on the events on the web pages, and once they found the event interesting, and they want to be one of the members, they have an option to participate in it, just list at the bottom of the webpage, there is a line with participate bottom shown all the time through browsing the activity introduction page.

4. Rating the events. This function is only open to the people who have participated in this activities. After attending the event, they will have a chance to review the events, is it great or bad? The criterion of participation is decided by the activities organizer when people attend the event, the organizer have a feature to confirm the attendance, once confirmed, the window of the participator will have the review bottom shown up, this time, those confirmed attendances are allowed to leave a comment to the activity.

The rating system is like this, five stars for perfect, and one start stands for the worst, and from one star to five star which represent from worst to awesome. What’s more, the user could also add some comments to it at the input box below.

4. Organise event. This function means every user is able to organise their own events. When the user starts the organizing, the user will have to fill the events form. Like post a listing apartment information on the Airbnb. First the organiser will have to upload at least one cover image, activity description including specific event location, events accurate time, requirements for potential participants, fee of the event. Former rating and reviews, if any.

5. Searching hobbies, the user will have the option to search hobbies at the main page, user can search hobbies like basketball or table tennis, once they found one, the result will **show up in a new page, showing the hobby main page??? Or just a list of results.**

Hobbie if there was nothing out shown in the result, the user could build their own hobby world, they will be offered an option like the github, if you didn’t find that branch, you can create it, so over here, we have that kind of function too, it could be to create my own hobby world, once the user click on that, the page will pop up a new page, which includes hobby name, hobby description(optional), after this, people will continue to the next step, save and create, after this, a new hobby world is created. And people could do what they want in that hobby world, whether to hold events or participate into the events other created.

6. Invent people to join the event. This means when a user participate in a particular event, and he/she wants more his/her friends to go with him/her, that user could click the button on the events introduction page and

**concern:9, Searching people by experience levels**

**10, Searching people**

7. Accept invitation or decline it, the people who has already been sign in our website just click on that invitation link and log in, they will see the activity page directly, and they can click on the join button, if the people who receive the invitation, and they are not a user, they have to sign in first, and then they will jump to the activity page then just the same as former action. The decline is just to ignore the invitation link.

8. Could browse any hobby community by locations, includes locate where the user is.

This is a function which could allow the user to set the current location on the web page which stood at the front right conor all the time, it is like the amazon to set up your current site, whether in America or England, and once you set up, you will have a option to find the events around you, the events around you will be popped to the front, not randomly placed, this time it is based on the distance, placed from nearest to the farest. And the user could also change their location later as they want.

9. User can check their registered events, by clicking on the button on the top right, inside a profile picture, the web page will pop up a list of options automatically when the mouse is put on it, the options are “My profile”, “My events”,“Log out”from top to bottom. When the user click on the bottom“My events”, then the web page go to a new page, which shows “the events you organized”, and “the events you registered”, the events are listed by time, from the lastest to farest, from top to the bottom in both the events you organized”and the events you registered” category

10. …..

5.2.2 Project manager

1. The managers take over the whole application, and can access to every part of the application. He is able to delete or fold unreasonable comments towards the events and leave a message to the every user. Like system notification.

2. The application meet all specified requirements from stakeholders.

3. Cost and development schedule should be done by the expected closing date.

5.2.3 Developers

1. The application structure should be clear and not ambiguity.

2. The application is easy to maintenance.

3. The development process is documented and traceable, technics can maintenance normally.

5. The development plan is complete and accurate.

5.2.4 Sponsors

1. The application can make profit within a fixed time.

2. The application will be delivered less or equal by the fixed time, in order to save more money and make profit early.

5.2.5 Advertisers

1. The advertisers need an image advertisement spot in striking place.

2. That advertisement picture should stay on the top of the main page for at least 3 seconds for each user.

3. The advertisers want this application to have some specific constraints on the image size and its contents and advertising time and frequency.

5.3 Business Requirements

5.3.1 Competitors

1. XXXXX

XXXXXX

2. XXXXX

XXXX

3. XXXX

XXXXX

5.3.2 Business Goal

1. This website could make profit within the first year.

2. The prototype could be delivered in three months with top 10 requirements in function features.

3. The website could serve over 3000 thousand people at the same time at least in the first period, and able to update the service within 2 days.

4. The system must have a capability to resist cyber attack or any kind of hacks.

5. The system must secure enough, will not leak user personal information.

5.3.3 Other requirements

1. Dimension Constraint Driver Degree of Freedom Cost up to 10% overrun from the initial estimate.

2. Features For Version 1.0, All required features must be fully operational and tested.

3. Quality For Version 1.0, All major features must be fully tested.

4. Failing allowance reaches at most 5% of user acceptance tests.

5. Schedule Version 1.0 must be delivered in 3 or 4 months.

6. Staff 6 staff work with full time for duration.

5.4 Others

1. Multiple language support

In our application, we will support English, simplified Chinese, Spanish

1. In case of hacking automatically data backup will be applied

In case of emergency, the system needs to backup necessary data once a week

6. Key Use Cases

6.1 Registration

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and Name:** | UC01(Sign up) | | |
| **Created By:** | Zhe Sun | **Date Created:** | 2/16/2019 |
| **Primary Actor:** | Guest(Non-registered user) | **Secondary Actors:** | None |
| **Description:** | Guests need to sign up with their valid email address and input password twice including one time password confirmation. Then system will verify the email address by sending an activation code. If the user receive the activation email and input the correct activation code, then, the signup process is successed. | | |
| **Trigger:** | Only the main page is able for guest to see, but for any part of the main page, guest(never sign up before) want to know more detail about that part, the guest has to sign up first. | | |
| **Preconditions:** | The guest could only have a brief look at the main page, if the guest want to know more detail information about Event, Hobby community, front page Hobby introduction salon | | |
| **Postconditions:** | Users are able to enter any part of the main page, they could check events information, join events, check hobby community and find their own, subscribe their own hobby communities, in another word, once sign up, the user could use all the function on the website. | | |
| **Normal Flow:** | 1. The guest clicks any button on the main page without logging in’.  2. The main page pops up an active window which shows “Sign up” and “Log in” small button on the upper right corner  3. The guest has to input a valid email address, password twice, then click “start” button at the middle bottom  4. The window will show another another small window in the same position on the website, which present a input box for user to input the activation code, at the same time, the system will send the guest an activation email which comprise an activation code,  5. The guest input the confirmation code correctly and click “Sign up” button at the bottom  6. The guest has already sign up successfully, and the website will refresh to a state which user has already logged in, and it will jump to the main page with login status. | | |
| **Alternative Flows:** | The first step can be triggered when guests click on ‘Sign Up’ which is on the upper right corner. | | |
| **Exceptions:** | 1, The guest typed in an invalid email address.  2, The guest input the wrong activation code. Either they check their email again within 10 minutes, or click on the “Send again” button down right the activation code input box, and receive a new activation code, then reinout it.  3, Password is too simple didn’t meet the type requirement. | | |
| **Business Rules:** | No user information leak | | |
| **Other Information:** | Password restriction: include both lower and upper case characters, at least one number or symbol, at least 6 characters long, no more than 16 characters. | | |

6.2 User log in

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and Name:** | UC02 (User log in) | | |
| **Created By:** | Zhe Sun | **Date Created:** | 2/16/2019 |
| **Primary Actor:** | Registered User | **Secondary Actors:** | None |
| **Description:** | Users log into the system | | |
| **Trigger:** | When users want to know more detail information on the website or take any actions. Or the user just clicking on the ‘Sign in’ button | | |
| **Preconditions:** | The user already has an account but didn’t log in yet. | | |
| **Postconditions:** | Users are able to browse ang module shown on the website. | | |
| **Normal Flow:** | 1. The user clicks on the “logging in” button.  2. The main page pops a small window which asks the user to input email address and password  3. The user clicks on the “Log in” button.  4. User logins successfully, jump to the main page with login status. | | |
| **Alternative Flows:** | 1. When the user browsing the main page, and click on anywhere on the main page, the website will pop up a sign window, on the upper right corner, a “Log in” button there, 2. The user can click on that “Log in button” to log in. The process is same as the “Log in” process described above. | | |
| **Exceptions:** | 1. The user typed a wrong password.  2. The user forget the password.  3. The user typed an invalid email address. | | |
| **Other Information:** |  | | |
| **Assumptions:** |  | | |

6.3 User forgot password

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and Name:** | UC03 (User forgot password) | | |
| **Created By:** | Zhe Sun | **Date Created:** | 2/16/2019 |
| **Primary Actor:** | Registered User | **Secondary Actors:** |  |
| **Description:** | During login process, the user forgot the password, the user could get password back by clicking on the “forgot password” button, and the system will send user a password reset link, once finished password reset process, the user could log in with the new password. | | |
| **Trigger:** | Click on ‘Forgot password’ button. | | |
| **Preconditions:** | User wants to log in but forget the password and is not able to log in. | | |
| **Postconditions:** | The user could log in with the newly reseted password. | | |
| **Normal Flow:** | 1, User click on the ‘Forgot password’ on lower right corner below the password button..  2, User received an email sent from system, which comprise a link for user to reset their password..  3, Users click on that link, and the system will jump to a web page which ask user to enter a new password, down the password box, there is password confirmation box, so the user have to input the new password twice.  4, Once go through the password reset process successfully, the user is able to log in to the website with the new password. | | |
| **Alternative Flows:** |  | | |
| **Exceptions:** |  | | |
| **Priority:** | Medium | | |
| **Other information:** | The newly reset password is not allowed to be the same as before, only brand new password. | | |

6.4 Input personal information

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and Name:** | UC04(Input personal information) | | |
| **Created By:** | Team 4 | **Date Created:** | 11/7/2018 |
| **Primary Actor:** | User | **Secondary Actors:** |  |
| **Description:** | Before publishing apartment as a tenant or searching roommates, user need to fill in information at this step. Some are required information such as gender, date of birth, staff or student grade. Some are optional personal information such as hobbies, schedule which shows to others. Personal information will be present in customize tags. | | |
| **Trigger:** | User click on ‘Looking for apartment’ button on homepage or ‘I am a roommate’ button on apartment publish page without finish this step. | | |
| **Preconditions:** | User have already logged in. | | |
| **Postconditions:** | User can get into ‘Looking for apartment’ page and publish the apartment information as a tenant. User’s tags can be showed by others. | | |
| **Normal Flow:** | 1, User get into the ‘Input personal information’ page  2, User input all the required information(gender, date of birth, staff or student grade) .  3, User can choose some tags that can describe himself.  4, User can customize tags.  5, User submit the information. | | |
| **Alternative Flows:** | At step 3 and 4, user can choose no tag, and when user submits the information there will be a suggestion: ‘some tags of you can help you match roommates easier.’ | | |
| **Exceptions:** | User input the invalid required information. | | |
| **Priority:** | Medium | | |
| **Other Information:** |  | | |
| **Assumptions:** |  | | |

6.5 Search apartment

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and Name:** | UC05(Search apartment) | | |
| **Created By:** | Team 4 | **Date Created:** | 11/7/2018 |
| **Primary Actor:** | Registered User | **Secondary Actors:** | Background auditors |
| **Description:** | User can use the location to search the apartments, view the information about apartment and add the apartment to the favorite list. | | |
| **Trigger:** | User clicks ‘Search apartment’ button. | | |
| **Preconditions:** | User completed ‘input personal information’ step. | | |
| **Postconditions:** | User can view the information of the apartment which meet the constraints and pull the apartment into favorite list if like. | | |
| **Normal Flow:** | 1, After users click the ‘search apartment’ button, user will jump to the ‘search’ page.  2, Users fill the location wanted to live.  3, After clicked ‘Search’ button, the page will show up a list of apartment choices on the right screen of the website.  4, User click the specific apartment can get into the page with detail information of the apartment such as virtual map pictures  5, User click ‘like’ button to add the apartment to the favorite list.  6, User click ‘find roommates’ button and jump to search roommates page. | | |
| **Alternative Flows:** | At step 2, if user didn’t input any location, system will show all the apartment information.  At step 6, if user doesn’t want to find roommates now, they can click ‘back’ button to go back to the apartments list page. | | |
| **Exceptions:** | The location user insert is invalid.  User didn’t finish the ‘input personal information’ step | | |
| **Priority:** | High | | |
| **Other Information:** |  | | |
| **Assumptions:** |  | | |

6.6 Find roommates

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and Name:** | UC06(Find roommates) | | |
| **Created By:** | Team 4 | **Date Created:** | 11/7/2018 |
| **Primary Actor:** | Registered User | **Secondary Actors:** | Background auditors |
| **Description:** | User can search roommate by gender, date of birth, student grade even personal tags. After searching, system will show a list of potential roommates, user can see the detail information and send friend invitation. If invited successfully, user can add potential roommates to a group with an apartment information. | | |
| **Trigger:** | User click ‘Find roommates’ button on the homepage or on the bottom of the apartment information page. User click ‘I am a roommate’ button when publish apartment. | | |
| **Preconditions:** | User completed ‘input personal information’ step. | | |
| **Postconditions:** | User have a list of potential roommates in ‘My profile’ and can send messages with potential roommates as a group. | | |
| **Normal Flow:** | 1, User input the search condition.  2, User get a list of potential roommates.  3, User open a specific potential roommate’s own page and check tags and brief introduction.  4, User click ‘invite’ button.  5, User can check whether others agree with the invitation in ‘My Profile/potential roommates’  6, User select one or multiple potential roommates to a group | | |
| **Alternative Flows:** | At step 1, User can input no condition to search all potential roommates.  At step 3, User can select multiple people to sent invitation without checking the detail information.  At step 6, if others refuse the invitation the user will see a ‘fail’ mark behind the potential roommate information. | | |
| **Exceptions:** | 1, User regret to send the invitation.  2, User create a group without apartment information  3, The apartment information of the group has been expired.  4, User search a potential roommate which has already lived in an apartment but forget to change the statue. | | |
| **Priority:** | High | | |
| **Other Information:** |  | | |
| **Assumptions:** |  | | |

6.7 My profile

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and Name:** | UC07(My profile) | | |
| **Created By:** | Team 4 | **Date Created:** | 11/7/2018 |
| **Primary Actor:** | User | **Secondary Actors:** |  |
| **Description:** | Users can check and edit the information about the apartment they published. Users can create a group or manage the member in a group. Users can manage messages and invitations in this step. | | |
| **Trigger:** | User click the ‘My Profile’ button on the homepage. | | |
| **Preconditions:** | User have already logged in. | | |
| **Postconditions:** | Main page for this application shows up. | | |
| **Normal Flow:** | (**Check and edit information about published apartment**)  1.1, Users click ‘My publish Management’  1.2, Users click one specific publishing information and check the information about the Apartment.  1.3, Users click ‘Edit’  1.4, After editing, users click save to change the information published.  (**Create a group and add members**)   * 1. Users click ‘My apartment Management’   2. Users click ‘Create a group’ then jump to the create group page   3. Users insert the group name and choose an apartment from the favorite list, then add group member to the group from potential roommates list.   4. Users click ‘Create’ to create a group and add all the members selected.   (**Manage invitations**)  3.1, Users click ‘Information Management’ and jump to a new page shows all the invitations.  3.2, Users click ‘Accept’ to add the inviter to the potential roommates list. | | |
| **Alternative Flows:** | At step 1.3, User can click ‘Cancel’ to cancel this edit without saving.  At step 2.3, User can add no roommate but can add roommates from later process.  At step 3.2, User can click ‘Refuse’ to refuse the invitations from others. | | |
| **Exceptions:** |  | | |
| **Priority:** | Medium | | |
| **Other Information:** |  | | |
| **Assumptions:** |  | | |

7. Quality Attributes

7.1 Usability

1. User can get command of using this application within 10 minutes.  
2. The application will have a build-in online help system to help user who run in a lost situation..  
3. When user want to use one specific function, they will experience no more than three times’ failure.   
4. User will be given a chance to re-edit their personal profile like detail information, but gender is not allowed to reedit after the first time input.

7.2 Performance

1. Response time is no more than 22ms when user click on any bottom.  
2, Throughput is more than 15000  
3. Concurrent users is more than 10000

7.3 Security

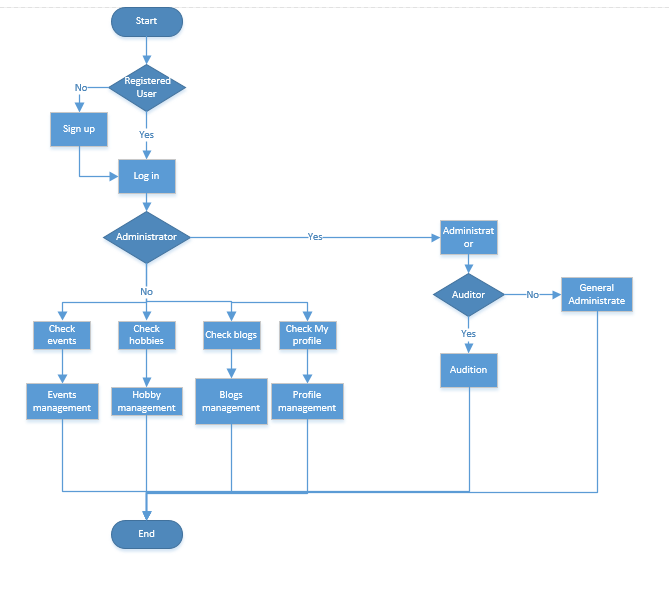
1. As Duck Home application for college wide students, this application still need comprehensive and detailed considerations in information security and user authentication to ensure that internal information is not leaked.  
2. The security of the application needs to meet the two OS systems of IOS and Android. This requires the vendor to fully consider the use of a set of security mechanisms to cover two systems, reducing the amount of re-development.  
3. As a support system for information release, network access security and permission security settings are required. For the published information content, a strict audit mechanism and filtering mechanism are required to avoid information security errors.

7.4 Reliability

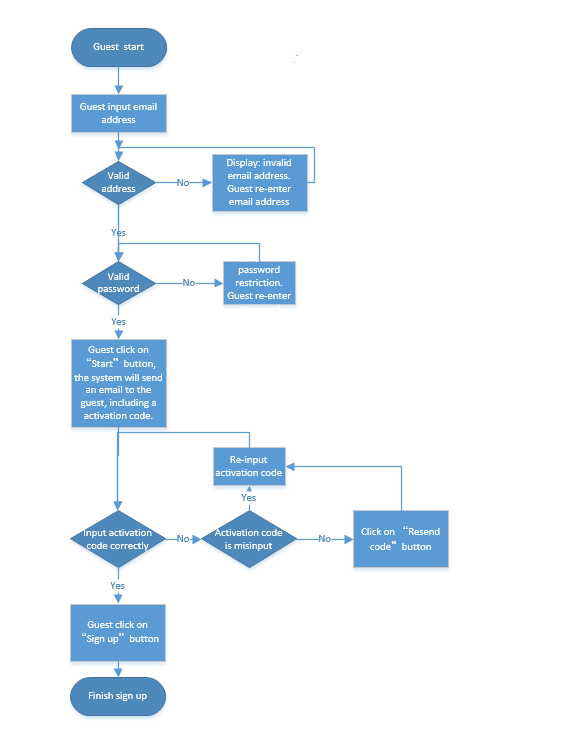
1. System failure rate is less than 1%.  
2. System availability is more than 99.99%.  
3. Server failure recovery time is less than 30 minutes.  
4. Database lost, the time about database recovery is no more than one hour.

8. Flow Charts

8.1 Overall

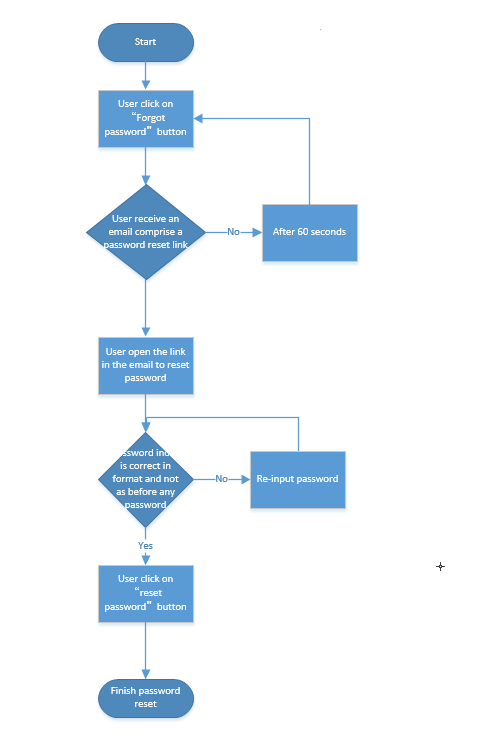


8.2 Register



8.3 Login

8.4 Forgot password/Reset password



8.4 Publish Events

8.5 Join Event

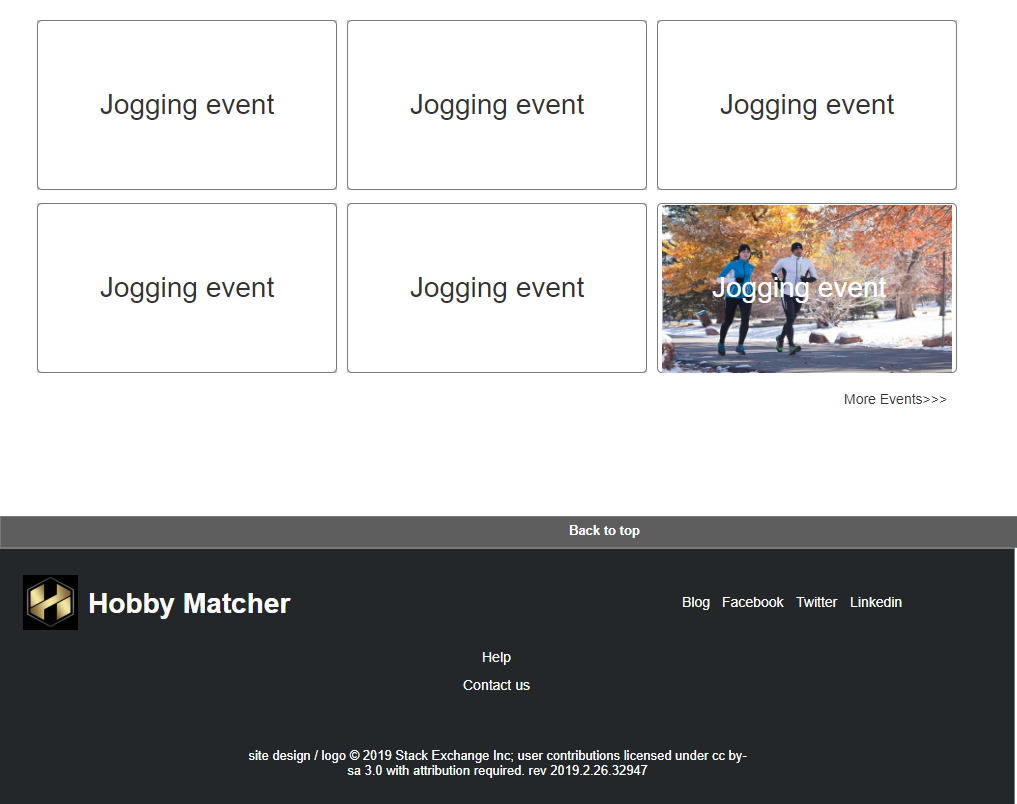
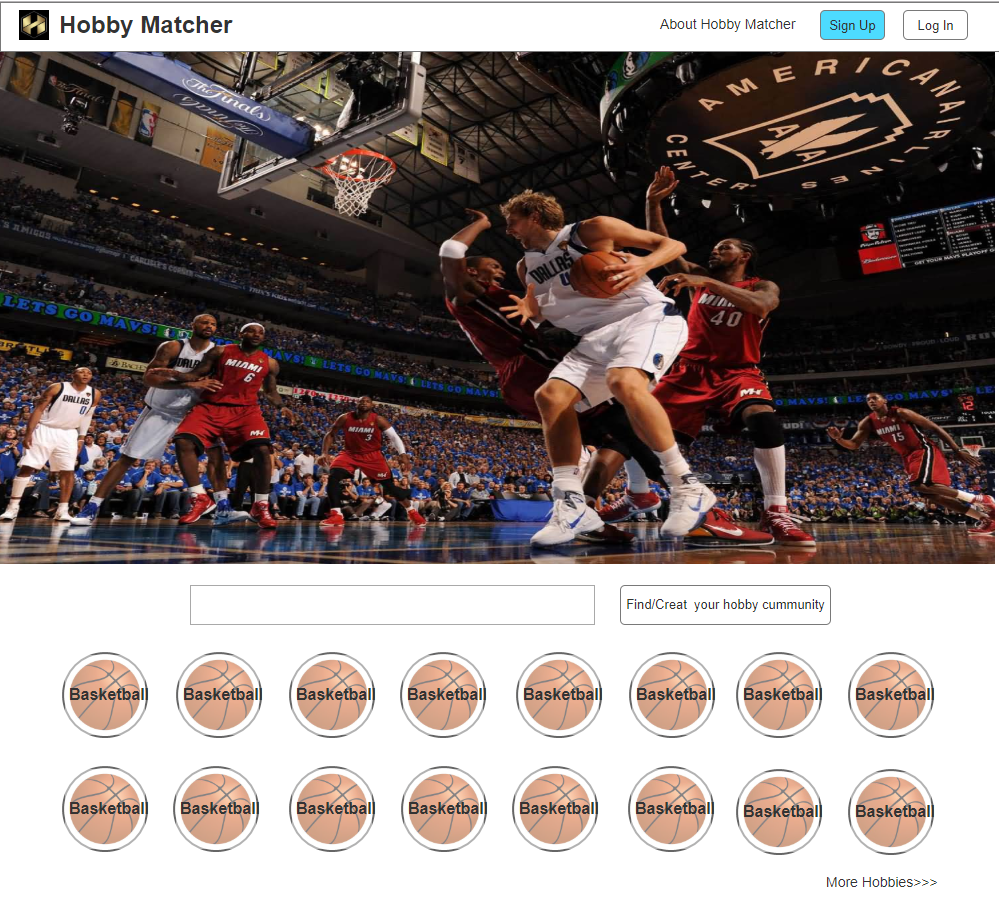
8.6 Create Hobby

8.7 Post Blog

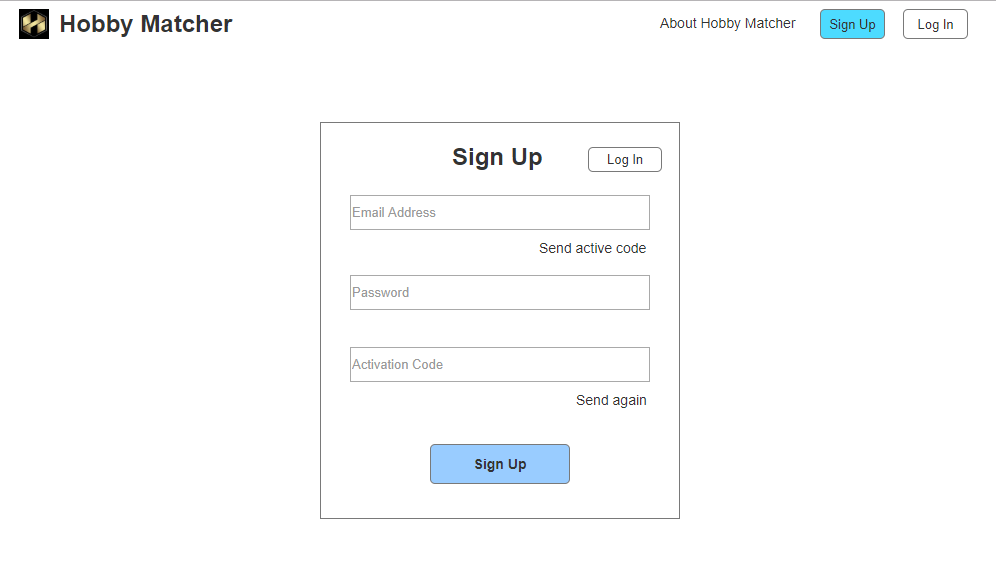
8.8 Comment Blog

9. UI Prototypes

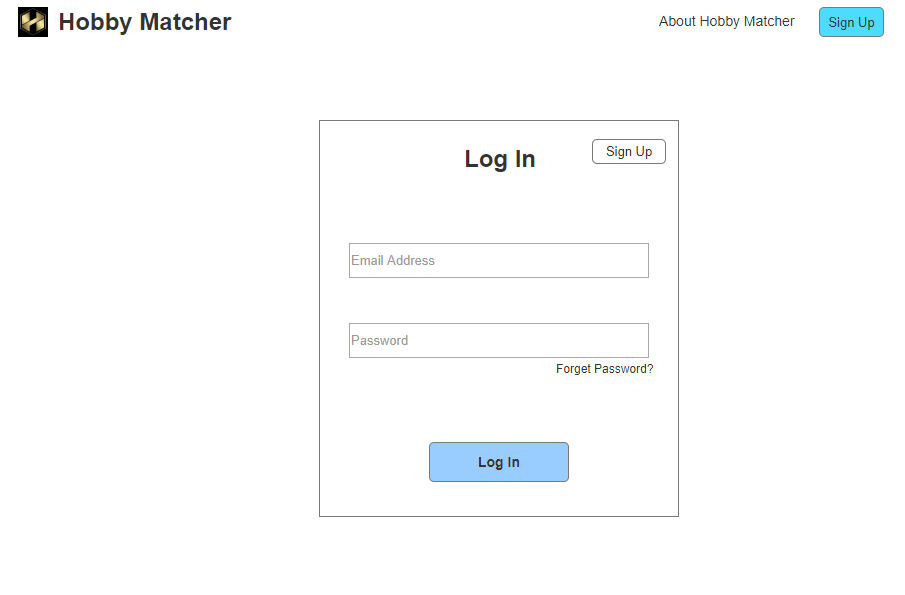
9.1 Homepage



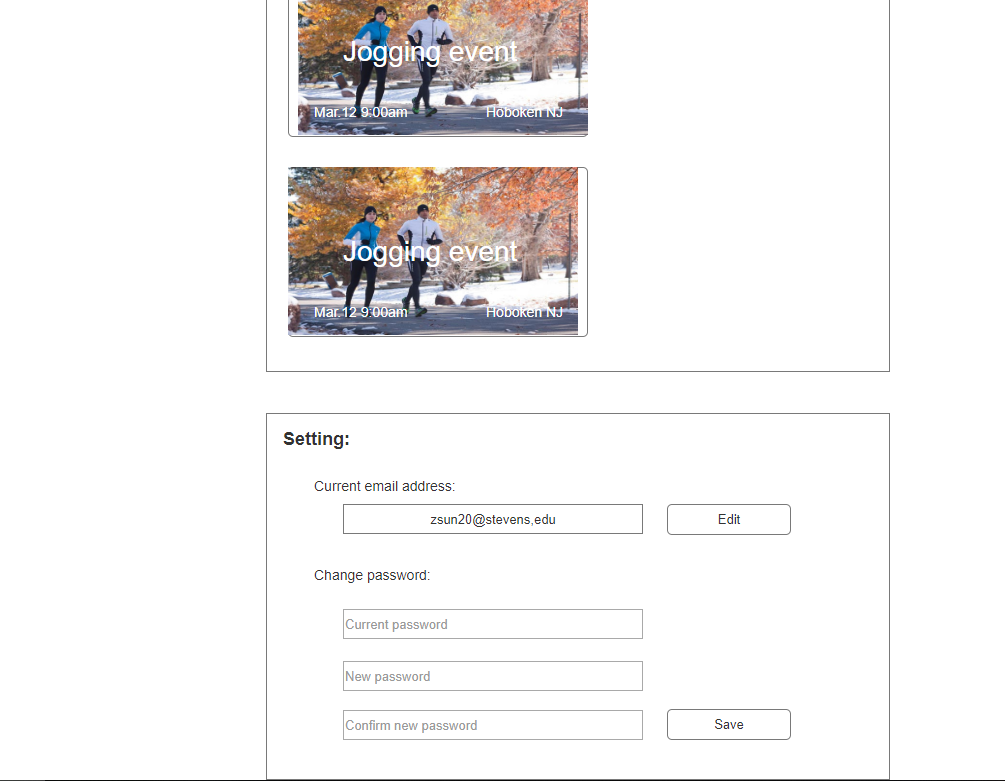
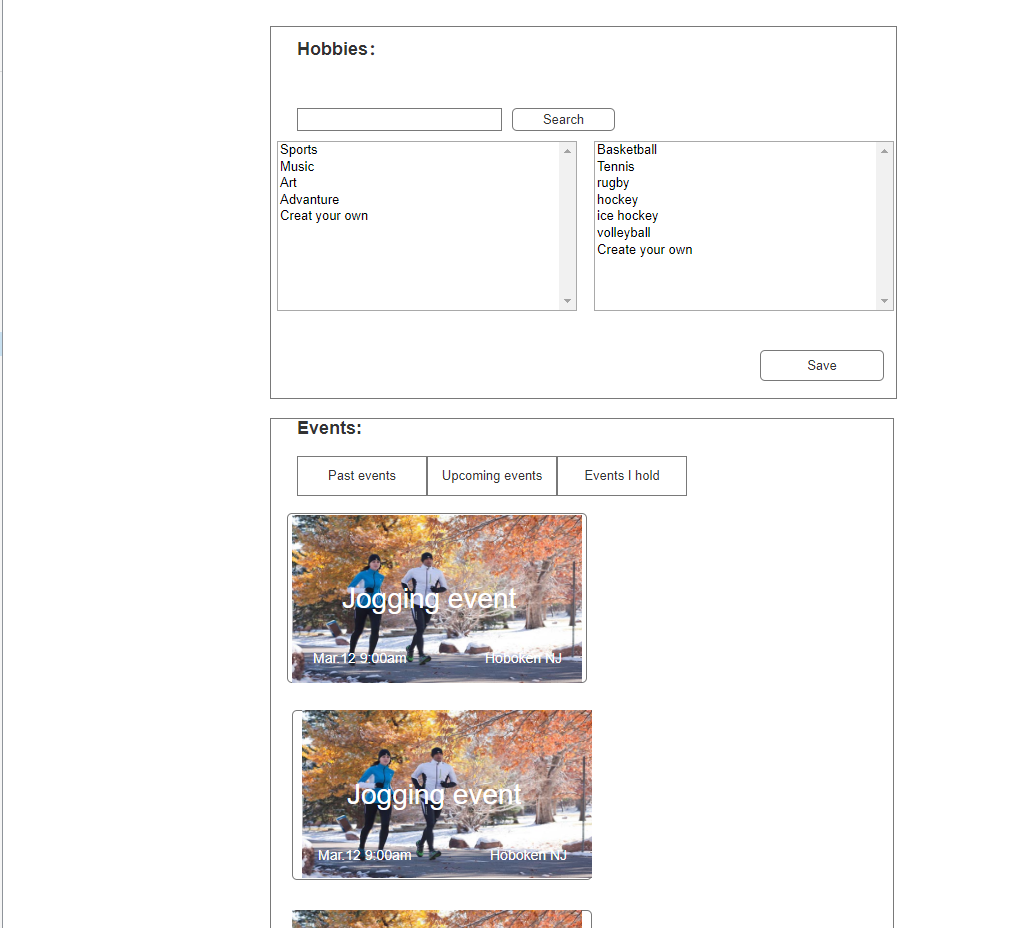
9.2 Sign up



9.3 Login



9.4 User profile



9.5 Publish Event

9.6 Join Event

9.7 Create Hobby

9.8 Post Blog

9.9 Comment Blog

9.10 System Message