Sprint 4 Backlog

Team 1

### **Profile page**

* Scenario: User can look at others' profiles
* Acceptance Test:
  1. Given a user is signed in
  2. When the user clicks another user's profile picture.
  3. Then the user is redirected to the person's profile page.
  4. The user can look at the person's detailed information including username, gender, age(depending on the users), etc.
* Development Requirements
  1. Implement frontend (include calendar display)
  2. Implement requests and responses in the front and the back
* Estimated time required: 25 hr

### **Profile Edit**

* Scenario: Users can edit their profiles
* Acceptance Test:
  1. Given the user is on own “Profile” page
  2. When the user clicks the “Edit” button
  3. Then the user is redirected to “Profile Edit” page
  4. When the user clicks “Save Changes” after changing his/her profile
  5. Then the user is redirected back to the “Profile” page with updated information.
* Development Requirements
  1. Implement frontend
  2. Implement requests and responses in the front and the back
* Estimated time required: 10 hr

### **Image file upload**

* Scenario: User uploads his/her profile picture
* Acceptance Test:
  1. Given the user is on the “Profile Edit” page
  2. When the user clicks the “Upload Profile Picture” button
  3. Then the “File Open” window comes up.
  4. When the user selects the picture and clicks “Open”
  5. Then the image file is uploaded to the server.
  6. And the user can see his/her profile picture through the browser.
* Development Requirements
  1. Frontend file selection
  2. File transfer using network protocol
  3. Backend API for accessing pictures
  4. Implement match thumbnail image upload
* Estimated time required: 15 hr

### **Interest-based recommendation**

* Scenario: A user wants to get recommendations of Matches he may be interested in
* Acceptance Test:
  1. Given the user is on the “Profile” page
  2. When the user clicks “Edit Interests” button
  3. And the user selects interests
  4. Then the user can get recommendations based on his interests in the home page
* Development Requirements
  1. Frontend sign-in page
  2. Frontend user profile page (display/edit)
  3. Backend match recommendation API
* Estimated time required: 20 hr

### **ML-based auto-classification**

* Scenario: A user wants an auto-classification when he creates a Match
* Acceptance Test:
  1. Given the user is on the “Create a Match” page
  2. When the user fills out the title form
  3. Then the natural language processing system classifies category, set location and time automatically.
* Development Requirements
  1. Choose what model to use.
  2. Download the model and check if it runs as expected
  3. Add into the project.
  4. Coordinate between frontend and backend
* Estimated time required: 30 hr

### **Google Map API**

* Scenario: Host selects the location using Google Map API
* Acceptance Test:
  1. Given the user is host of one matching
  2. And the user is on the match create/edit page of the match.
  3. When the user clicks Google Map Icon
  4. Then Google Map API Select Location window will pop up
  5. When the user selects the location in the API window
  6. And the user clicks Select button
  7. Then the location is set to the venue of the event
* Development Requirements
  1. Register Matchmaker into Google API service
  2. Coordinate in frontend
* Estimated time required: 25 hr

### **Chatroom API**

* Scenario: Participants can talk with each other using a Chatroom API
* Acceptance Test:
  1. Given a user is the host or participants of a match
  2. And the user is on the chatroom page of the match
  3. When the user types messages on the Chatroom API window
  4. Then the message is sent to the chatroom
* Development Requirements
  1. Find an appropriate Chatroom API
  2. Install the API in the project
  3. Coordinate in frontend
* Estimated time required: 25 hr

### **Notice by host**

* Scenario: Host posts new notice about his/her matching
* Acceptance Test:
  1. Given the user is host of one matching
  2. And the user is on the detail page of the match.
  3. When the user clicks the ‘Notice’ button
  4. Then ‘Add Notice’ window will pop up
  5. When the user fills new notification in the input box
  6. And the user clicks ‘Notice’ button
  7. Then new notification is added to matching detail page
  8. And all attendees of this matching get a notification of it.
* Development Requirements
  1. Frontend notice creation page for hosts
  2. Backend model create
  3. Implement communication between the front and back
  4. Implement Notification alert in the header
* Estimated time required: 10 hr

### **Research/Investigate tasks**

* Profile Image Upload - How to store images in backend
* API usage (Google Map / Chatting)
* NLP models
* Calendar display