Requirements specifications

User requirements

1) Users must be able to add other people to their friend list and chat with them only after they have met them.
2) Users must be able to filter out events and places. (E.g. if they don’t want to go to a pub, then they must be able to opt out of pub suggestions)
3) Users must be able to “opt out” from the matching system and just keep using the platform to organise their schedule.
4) Users must be able to delete their account if they don’t want to use the platform anymore.
5) Users must be able to upload pictures of themselves and have a photo gallery.
6) Users must be able to filter users that they want to get match with based on things such as hobbies, locality, schedule, course and age.
7) After starting a conversation with a friend within the application, users must be able to view the entire message history.
8) Businesses must be able to identify themselves during the sign-up process using their business number.
9) Businesses must be able to promote events on the application.
10) Users must be able to make their photo gallery private.
11) Users must be able to make their profiles private.
12) Users must be able to easily tell all the other users who are going to attend an event if they had a last-minute problem and won’t attend the event anymore.
13) Users must be able to easily “tell” the system whether they are going to attend a certain event or not.
14) If users don’t find their hobbies inside the drop-down list, they must be able to add them. *[Ref 22]*
15) The user must be able to add family members or friends to a trustee list (this will be later used when the “Emergency!” is pressed).
16) The user must be able to call or text other users who are going to attend his/her same events.

System requirements

Functional requirements

17) The system must verify each user’s university email address before letting them use the application.
18) The system must check that users submitted a valid ID (e.g. passport, driving licence etc…) before letting them access the application services.
19) Once the system receives the valid ID, it must verify the user’s age and identity. *[Ref 18]*
20) If the user who is currently signing up is older than 30 years old, the system must notify and tell him/her how many users around his/her age are active on the platform.
21) After the user has signed up, the system must ask him/her to enter his/her personal data.
22) After the user has signed up, the system must ask him/her to select his/her hobbies from a drop-down list.
23) The system must normalize the data entered by the user during the sign-up process.
24) Once a user has added a hobby, the system must add it to the hobbies database. *[Ref 14]*
25) Once the system added a hobby to the hobbies database, the algorithm must not take it into account for future user matches until it is approved by a human. *[Ref 24]*

26) Each time users go to an event, the points system must give them 20 points.
27) If a user accepts to go to an event but he/she cancels it at last minute, the system must detract him/her 10 points.
28) If a user gets reported for a valid reason, the system must detract him/her 20 points.
29) The users' data must be sent and stored in a cloud database storage.
30) The algorithm must try to match users on as many "factors" as possible (factors in order of importance are: course, university, hobbies, location and age).
31) The "Emergency" button must immediately send a custom text message to people who are members of the user's trustee list.
32) If a user presses the "Call me" or "Text me" button (the one next to each user who is going to attend the user's same event), the system must call the selected user or open the messages application in order for the user to send him/her a message.
33) Whenever users want to exchange their points for discounts, the system must first check if they have enough to do so.
34) Whenever users exchange their points for discounts, the system must detract from their total amount of points, the points that they have just spent.
35) The algorithm must notify the users who got matched.
36) The system must remind users to update their weekly schedule each week.
37) Whenever the user updates the schedule/calendar, the system must auto-save the changes the user has made.
38) The system must never store or share a user's live location unless the "Emergency!" button is voluntarily pressed by the user.
39) The system must verify that a business entered a valid business number and has supplied at least 10 discount codes (which will later be available to users who go to their events through our application) before letting it promote its events on the application. *[ref 9]*

Non-functional requirements

40) Services provided by the system shall "talk" between each other using APIs.
41) The system shall be developed using AWS service.
42) The system must achieve 99% up time.
43) The precision of calculations with derived data shall be at the same degree of precision as the originating source data.
44) A maintenance developer shall test the system's performance every 2 weeks in order to be sure that all services provided by the application are working properly.
45) The system shall be developed in such a way that the performance of the application should not degrade with an increase in the number of services offered.
46) The system should be able to handle a minimum of 2000 users simultaneously.
47) The system must have a high response time (at least less than 2ms).
48) The matching algorithm must achieve 99% accuracy.
49) The users' data management must run on a SQL database.
50) Exchanging points for discounts should be achievable in less than 2 clicks.
51) Each service provided by the application should be accessible in less than 3 clicks.

Usability requirements

52) The calendar/schedule should be intuitive to use for the user.
53) The chatting system should have a simple but efficient interface.

54) People with no training and no understanding of English shall be able to use the application.
55) The map displaying details of a matched user’s event should be simple and easily readable.
56) The “Help Me!” button within the “My Map” section (the one where all the details regarding the event a user is going to attend are displayed) must be apparent and noticeable.
57) Each main page within the application must be accessible through a tab navigation.