H.I.P.S.T.E.R

Humans in Programming, Science, Technology, Engineering, Research

Use Cases

Authors

*Joe M' Deon, Stephen Gilardi, Sara Farsi, Saveen Chadalawada, Kann Chea,
Vincent Yu, Sher Zahed, Francisco Aceves, Brandon Powers, & Mikey Cho*

Table of Contents

Legend	
[UC-SXX]	Use Case - Social
[UC-PXX]	Use Case - Profile
[UC-GXX]	Use Case - Group
[UC-SCXX]	Use Case - Schedule

🟥 **INCOMPLETE**

🟩 **COMPLETE**

UC-S01: See List of Friends

Description: The user shall be able to see a list of the user’s existing friends.

User Goals: The user wants to see a list of friends to be able to delete or organize his/her list.

Dependency Use Cases: Add/Remove Friends (UC-S02)

Details:

Priority: 2

Progress Status: Incomplete

Test Phase Status: Planned

Pre-Conditions:

1. User has added friends.
2. If the user does not have any friends, an empty list is shown.

Post-Conditions:

1. User shall see a list of his/her friends, ordered alphabetically.
2. The user shall see a “delete” option next to all of his/her friends’ names.

Trigger: User has clicked on the button to “View Friends”

Workflow:

1. The user is in Profile View.
2. The user clicks on the “friends” button.
3. The system accesses the database and displays a list of the user’s friends.

Requirements:

1. The standard user shall be able to see a list of friends.

UC-S02: Add Friend

Description: The user shall be able to add friends using their email address/username.

User Goals: The user wants to add a friend to his/her list, so he/she can find the friend using their friend's profile name instead of email address.

Dependency Use Cases: Create an Account (UC-P01), Search a Friend (UC-S05)

Details:

Priority: 2

Progress Status: Incomplete

Test Phase Status: Planned

Pre-Conditions:

1. User shall type the email address of the friend she/he wish to add.
2. User shall verify if the profile found is the correct one.
3. User shall click on "add" button.

Post-Conditions:

1. A notification is sent to the friend.

Trigger: User has found the friend using email address, then user clicks on the "add" button.

Workflow:

1. The user shall be in the "profile" view.
2. The user shall click on "add friend" button.
3. The system shall display an input form.
4. The user shall input the email of their desired friend.
5. The user shall submit.
6. The system shall search the database for the user with that email.
7. The system shall add that user to the user's friend list.
8. The system shall display an alert notifying the user that the friend has been added.

Requirements:

1. The standard user shall be able to add a friend to their friends list using the friend's email address.
2. The system shall verify if the entered email matches any of the accounts.

UC-S03: Remove Friends

Description: The user shall be able to remove friends from the user's Friends List.

User Goals: The user wants to remove a friend if he/she does not wish to have them as friends.

Dependency Use Cases: See List of Friends (UC-S01), Search For a Friend (UC-S05)

Details:

Priority: 2
Progress Status: Incomplete
Test Phase Status: Planned

Pre-Conditions:

1. User is in the "My Friends" view.

Post-Conditions:

1. If the user chose to remove a friend, that friend shall no longer appear in list.

Trigger: User has clicked on the "friends" button, then the user clicks on "remove" button.

Workflow:

1. The user is in profile view.

path1:

2. The user shall click on the "friends" button
3. The system shall access the database and display a list of the user's friends.
4. The user shall scroll to the friend and click on the "remove" button next to the friend's name.
5. The system shall display a safety check to confirm the user's remove choice.
6. The user shall confirm their action.
7. The system shall remove the friend from the user's friend list in the database.

path2:

2. The user shall click on the "friends" button.
3. The system shall access the database and display a list of the user's friends.
4. The user shall input the friend's name into the search bar at the top of the list.
5. The system shall display a reduced list of friends containing the name searched.
6. The user shall click the "remove" button next to the friend to be removed.
7. The system shall display a safety check to confirm the user's remove choice.
8. The user shall confirm their action.
9. The systems shall remove the friend from the user's friend list in the database.

Requirements:

1. The system shall require verification to delete a friend from a user's friend list
2. The standard user shall be able to remove a friend

UC-S04: Chat Box

Description: The user shall be able to chat with group members via an accessible chat box specific to a group they are a member of.

User Goals: The user wants to have a chat box for groups to make communication easy and have available meeting times visible next to the chat box.

Dependency Use Cases: Create a Group (UC-G01)

Details:

Priority: 2
Progress Status: Incomplete
Test Phase Status: Planned

Pre-Conditions:

1. User is a member of a group.
1. User is in the view for a particular group.

Post-Conditions:

1. User shall see a chat box after clicking on "chat" icon.

Trigger: User has clicked on the "chat" icon

Workflow:

1. User is in group view.
2. User clicks on chat icon.
3. System accesses database to retrieve chat history.
4. System presents chat box, which contains chat history.
5. User types into text field and submits message.
6. System updates chat history to reflect most recent correspondence.
7. User can close or minimize chat box.

Requirements:

1. The user shall be able to see a chat box
2. The user shall be able to enter texts in the chat box.
3. The system shall show all messages/texts sent to the chat box.

UC-S05: Search for a Friend

Description: The user shall be able to search through his/her friends and search for people in general with the username or profile name.

User Goals: The user wants to find people either in his/her list of friends or not in his/her list of friends.

Dependency Use Cases: Create User Account

Details:

Priority: 2
Progress Status: Incomplete
Test Phase Status: Planned

Pre-Conditions:

1. The user shall have an account.
2. The user shall know and type the email address of the person.

Post-Conditions:

1. The user sees the person’s name and profile picture he/she was searching for or “User Not Found”

Trigger: User clicks on “Search for Friend”.

Workflow:

1. User is on a page that has a “Search for Friend” button
2. User clicks on button
3. User enters a unique email address of the user being searched for
4. User submits
5. System searches database for user that matches unique email address
6. If found (assume only one result can exist), return result, else return “User Not Found”

Requirements:

1. The user shall be able to see a search box and be able to type an email.
2. The system shall query the database for a list of users matching the email string entered.

UC-P01: Create an Account

Description: The user shall be able to create an account to have a personalized experience with the website.

User Goals: The user can access all other functions of the web app after logging in to their account.

Dependency Use Cases: Unique Usernames (UC-P04)

Details:

Priority: 1
Progress Status: Complete
Test Phase Status: Planned

Pre-Conditions:

1. The user must know the login web address.

Post-Conditions:

1. The user shall be in profile view.
2. The user shall be able to insert schedule.
3. The user shall be able to create a group.

Trigger: The user uses his/her email address and a password to create an account. Then the user click on “Sign Up” button.

Workflow:

1. The user is at the login screen.
2. The user shall enter their email.
3. The user shall enter their password.
4. The user shall enter their password again to confirm.
5. The user shall click the sign-up button.
6. The system shall verify the user’s email is unique and that both passwords are the same and store the data in the database.
7. The system shall display the user’s new blank profile.

Requirements:

1. The system shall require a unique/valid username in the form of an email [UC-P05]

UC-P02: Login to the App

Description: The user shall be able to login to their account with their email and password.

User Goals: After creating an account, the user wants to login in to the app every time he/she wants to use it to be able to revise/edit schedule or add/remove a group.

Dependency Use Cases: Create an Account (UC-P01)

Details:

Priority: 1

Progress Status: Complete

Test Phase Status: Planned

Pre-Conditions:

1. The user shall have an account.
2. The
3. The System must be able to store user's information.

Post-Conditions:

1. The user shall be in profile view.

Trigger: The user must enter his/her username and password, then user must click on "Login" button.

Workflow:

1. The user is at the login screen.
2. The user shall enter their username.
3. The user shall enter their password.
4. The user shall click the login button.
5. The system shall verify the user has an account and log the user in.

Requirements:

1. The system shall request a valid username/password combination.
2. The standard user shall be able to login.

UC-P03: Forgot Password

Description: The user shall be able to reset/change password when password is forgotten.

User Goals: The user must want to recover a lost password.

Dependency Use Cases: Create an account(UC-P03)

Details:

Priority: 1

Progress Status: Incomplete

Test Phase Status: Planned

Pre-Conditions:

1. The user has an account.

Post-Conditions:

1. An email will be sent to the user for password reset.

Trigger: The user must click on “Forgot Password” button.

Workflow:

1. The user shall be in the login screen.
2. The user shall click the “forgot password” button.
3. The system shall display a screen where the user can input their email.
4. The user shall input their email address.
5. The system shall email a link to the user’s email.
6. The user shall click on the link in their email taking them to a forgot password screen.
7. The user shall input their new desired password in two input boxes and click confirm.
8. The system shall confirm that both boxes have the same password.
9. The system shall overwrite the old password with the new password in the user’s database.
10. The system shall display an alert notifying that the user has a new password.

Requirements:

1. The system shall require a valid username and verification code sent to them via email.
2. The standard user shall be able to “remember” their password.
3. The standard user shall be able to reset their password.

UC-P04: Change Account Profile Info

Description: The user shall be able to access and edit their profile settings to make it a better experience for themselves whenever they want.

User Goals: To edit certain profile friends.

Dependency Use Cases: Have an account (UC-P01)

Details:

Priority: 1

Progress Status: Incomplete

Test Phase Status: Planned

Pre-Conditions:

1. The user must have an account.

Post-Conditions:

1. The user has changed desired profile information.

Trigger: The user clicks the account information button.

Workflow:

1. The user is in profile view.
2. The user shall click on the account info button.
3. The system shall display all personal info.
4. The user shall click the edit button next to the item they want to change.
5. The system shall allow the user to edit certain info.
6. The user shall edit the information.
7. The user shall click the done button.
8. The system shall save the information to the database.

Requirements:

1. The user shall be able to access and edit settings.

UC-P05: Unique User Names

Description: The user shall have a unique username to login with that will most likely be their email.

User Goals: To create a distinct username.

Dependency Use Cases: Create an Account

Details:

Priority: 1
Progress Status: Incomplete
Test Phase Status: Planned

Pre-Conditions:

1. The user shall input a string representing their username.

Post-Conditions:

1. The system has distinct usernames for each user and the user can be identified uniquely by this username.

Trigger: User shall enter his/her email address as username.

Workflow:

1. The user is in the signup view.
2. The user attempts to enter a username.
3. The system checks if there are any other username that match the user's inputted string.

Path 1:(The string matches another username string)

1. The system requests the user to input their username again.

Path 2:(The string is unique)

1. The system accepts the string and stores it as the user's username.s

Requirements:

1. The system shall only accept usernames that are unique
2. The system shall identify users only by their unique username

UC-P06: Profile Name

Description: The user shall be able to create a profile name that is not necessarily unique and that will be shown in their groups.

User Goals: The user wants to be able to set a username that will serve as their default display name in groups.

Dependency Use Cases: Create an Account (UC P01)

Details:

Priority: 2

Progress Status: Incomplete

Test Phase Status: Planned

Pre-Conditions:

1. The user shall input a desired profile name.
2. The user shall be trying to create an account.

Post-Conditions:

1. The system shall have a profile name stored in the parse database for the user.
2. The system shall display the user's profile name in their profile view and in their groups unless otherwise set. (set group display name)

Trigger: The user the be setting click on the sign-up tab in the welcome screen.

Workflow:

1. The user is in the sign-up screen.
2. The user enters their desired profile name in the field labeled profile name.
3. The system stores the user's desired profile name in their database.
4. The user clicks submit.
5. The system displays a confirmation message.

Requirements:

1. The system shall store each members profile name without needing it to be unique
2. The standard user shall be able to choose a personalized profile name

UC-P07: Log Out

Description: The user shall be able to log out of the app at any time.

User Goals: The user wants to be able to set a username that will serve as their default display name in groups.

Dependency Use Cases: Create an Account (UC P01)

Details:

Priority: 2

Progress Status: Incomplete

Test Phase Status: Planned

Pre-Conditions:

1. The user shall input a desired profile name.
2. The user shall be trying to create an account.

Post-Conditions:

1. The system shall have a profile name stored in the parse database for the user.
2. The system shall display the user's profile name in their profile view and in their groups unless otherwise set. (set group display name)

Trigger: The user the be setting click on the sign-up tab in the welcome screen.

Workflow:

1. The user is in the sign-up screen.
2. The user enters their desired profile name in the field labeled profile name.
3. The system stores the user's desired profile name in their database.
4. The user clicks submit.
5. The system displays a confirmation message.

Requirements:

1. The standard user shall be able to log out of the application at any time.

UC-P08: Profile View

Description: The user shall be able to log out of the app at any time.

User Goals: The user wants to be able to set a username that will serve as their default display name in groups.

Dependency Use Cases: Create an Account (UC P01)

Details:

Priority: 2
Progress Status: Incomplete
Test Phase Status: Planned

Pre-Conditions:

3. The user shall input a desired profile name.
4. The user shall be trying to create an account.

Post-Conditions:

3. The system shall have a profile name stored in the parse database for the user.
4. The system shall display the user's profile name in their profile view and in their groups unless otherwise set. (set group display name)

Trigger: The user the be setting click on the sign-up tab in the welcome screen.

Workflow:

6. The user is in the sign-up screen.
7. The user enters their desired profile name in the field labeled profile name.
8. The system stores the user's desired profile name in their database.
9. The user clicks submit.
10. The system displays a confirmation message.

Requirements:

1. The Standard User shall be able to view their profile.

UC-G01: Create a Group
Description: The user shall be able to create a new group.
User Goals: The user wants to create a new group to see meeting times.
Dependency Use Cases: Create an account (UC P01), Login into the App (UC P02), Search for a Friend (UC S05)
Details: **Priority:** 1 **Progress Status:** Incomplete **Test Phase Status:** Planned
Pre-Conditions: 1. The user shall have an account. 2. The user shall click "add" group button. 3. The user shall input a group name.
Post-Conditions: 1. The system shall store the group in the parse database for the user.
Trigger: The user shall click on the "add group" button.
Workflow: 1. The user is in "Groups" view. 2. The user clicks on the "add group" button. 3. The system displays a popup with a group name field, an add friend field, and an add member field. 4. The user shall input a group name. 5. The system shall check that the group name is not already used by the user. 6. The user shall search for friends and add them to the group. 7. The system shall check that the friend is not already in the group. 8. The user shall add non-friends to the group by adding emails. 9. The system shall check that the person is not already in the group. 10. The system shall create a group in the parse database. 11. The system shall display a confirmation window.
Requirements: 1. The user shall be able to create a group. 2. The system shall make the user the leader of the group and able to edit group settings.

UC-G02: View Members of Groups I am a Part Of

Description: The user shall be able to view a comprehensive list of the group roster.

User Goals: The user shall gain information about who their groupmates are for a specific group.

Dependency Use Cases: Create an account (UC P01), Login into the App (UC P02)

Details:

Priority: 2

Progress Status: Incomplete

Test Phase Status: Planned

Pre-Conditions:

1. Be a member in the group

Post-Conditions:

1. The user will be able to see their groupmates for a specific group

Trigger: The user shall click on the “members” button in the specific group’s profile view

Workflow:

1. The user shall navigate from their profile screen to the group in question
2. The system shall display the group’s profile view
3. The user shall click on the “members” button for the group.
4. The system shall display the group leader followed by the group members.

Path 1: The user stays on the group info display view

5. The user shall click on a member to view info about them.

Path 2: The user leaves the group info display view

5. The system shall display the group’s profile view

Requirements:

1. The user shall be able to view his/her group
2. The system shall retrieve the names of the group members of the user’s selected group
3. The user shall be able to view the members in the specified group

UC-G03: View Details About Group Meetings

Description: The user shall be able to view details such as time, location, and the topic of the meeting.

User Goals: The user shall gain all the information about group meetings

Dependency Use Cases:

Details:

Priority: 2
Progress Status: Incomplete
Test Phase Status: Planned

Pre-Conditions:

1. The user must be a member of the group in question

Post-Conditions:

1. The system shall display all details pertaining to upcoming group meetings to the user.

Trigger: The user shall click on the "info" button in the specific group's profile view.

Workflow:

1. The user shall navigate from their profile screen to the group in question
2. The system shall display the group's profile view
3. The user shall click on the info button for the group.
4. The system shall display information about upcoming events in the group.

<u>Path 1:</u> The user stays on the group info display view

5. The user shall click on a member to view info about them.

<u>Path 2:</u> The user leaves the group info display view

5. The system shall display the group's profile view

Requirement:

1. The user shall be viewing the weekly schedule for the group
2. The system shall retrieve details of each event from the database and display them

UC-G04: View Groups I am In

Description: The user shall be able to see the groups in which they are a member of.

User Goals: The user shall be able to see a list of the groups he/she is a member of.

Dependency Use Cases:

Details:

Priority: 1

Progress Status: Incomplete

Test Phase Status: Planned

Pre-Conditions:

1. The user must be logged in.
2. The user must be a member in at least one group

Post-Conditions:

1. The system shall display the groups that the user is a member of.

Trigger: The user shall click on the "groups" tab from their profile view

Workflow:

1. The user shall begin in their profile view
2. The user shall click on the "groups" tab
3. The system shall display all the groups the user is a member of.

Path 1: The user shall click on a group

1. The system shall display the group's profile view

Path 2: The user shall click on the "profile" tab

1. The system shall display the user's profile

Requirements:

1. The user shall be logged in
2. The user shall be able to enter the groups tab/view
3. The system shall retrieve from the database the groups to which the user belongs
4. The system shall display the groups to the user

UC-G05: Simple Way to Find Good Meeting Times

Description: The user shall be able to choose a good meeting in a simple and efficient way.

User Goals: The system shall have a simple and intuitive way to find all meeting times over all the member's schedules.

Dependency Use Cases: Login to the App (UC-P02), Re-diff Schedules (UC-SC07), Providing Alternative Meeting Times (UC-SC08),

Details:

Priority: 1

Progress Status: Incomplete

Test Phase Status: Planned

Pre-Conditions:

1. The user must be able to choose a common meeting time between members provided by the system.

Post-Conditions:

1. The system shall display

Trigger: The user must have created an event within a group of members.

Workflow:

1. The user shall start in the profile view.
2. The user shall click on their groups.
3. The system shall display all the groups the user is in.
4. The user shall click on a group to enter the group's profile for which they are *group leader*.
5. The system shall display the group's profile.

Path 1: The user shall choose to add a new event.

6. The system shall display a form to create an event.
7. The user shall fill out this form and submit it.
8. The system shall display the best times to hold this event.
9. The user shall be able to select one or more to assign to the event.

Path 2: The user shall click on a pre-existing event to view it's details.

1. The system shall display a form with the event's details filled out.
2. The user shall be able to view the best times found by the system.
3. The user shall be able to edit this information and submit it again.

Requirements:

1. The user shall be creating/editing an event.
2. The user shall request for the best time to coordinate the event
3. The system shall access the schedules of all users in the group

4. The system shall use an algorithm to find the best times for the event
5. The system shall display the best times to the user

UC-G06: Create a Group Meeting

Description: The user shall be able to choose a good meeting in a simple and efficient way.

User Goals: The system shall have a simple and intuitive way to find all meeting times over all the member's schedules.

Dependency Use Cases: Login to the App (UC-P02), Re-diff Schedules (UC-SC07), Providing Alternative Meeting Times (UC-SC08),

Details:

Priority: 1
Progress Status: Incomplete
Test Phase Status: Planned

Pre-Conditions:

2. The user must be able to choose a common meeting time between members provided by the system.

Post-Conditions:

Trigger: The user must have created an event within a group of members.

Workflow:

10. The user shall start in the profile view.
11. The user shall click on their groups.
12. The system shall display all the groups the user is in.
13. The user shall click on a group to enter the group's profile for which they are *group leader*.
14. The system shall display the group's profile.

Path 1: The user shall choose to add a new event.

15. The system shall display a form to create an event.
16. The user shall fill out this form and submit it.
17. The system shall display the best times to hold this event.
18. The user shall be able to select one or more to assign to the event.

Path 2: The user shall click on a pre-existing event to view it's details.

4. The system shall display a form with the event's details filled out.
5. The user shall be able to view the best times found by the system.

6. The user shall be able to edit this information and submit it again.

Requirements:

1. The group leader shall be able to create a group meeting.

UC-G07: View single Group I am

Description: The user shall be able to choose a good meeting in a simple and efficient way.

User Goals: The system shall have a simple and intuitive way to find all meeting times over all the member’s schedules.

Dependency Use Cases: Login to the App (UC-P02), Re-diff Schedules (UC-SC07), Providing Alternative Meeting Times (UC-SC08),

Details:

Priority: 1

Progress Status: Incomplete

Test Phase Status: Planned

Pre-Conditions:

3. The user must be able to choose a common meeting time between members provided by the system.

Post-Conditions:

Trigger: The user must have created an event within a group of members.

Workflow:

19. The user shall start in the profile view.
20. The user shall click on their groups.
21. The system shall display all the groups the user is in.
22. The user shall click on a group to enter the group’s profile for which they are *group leader*.
23. The system shall display the group’s profile.

Path 1: The user shall choose to add a new event.

24. The system shall display a form to create an event.
25. The user shall fill out this form and submit it.

26. The system shall display the best times to hold this event.
27. The user shall be able to select one or more to assign to the event.

Path 2: The user shall click on a pre-existing event to view it's details.

7. The system shall display a form with the event's details filled out.
8. The user shall be able to view the best times found by the system.
9. The user shall be able to edit this information and submit it again.

Requirements:

1. The user shall be able to view a single group they are in.

UC-G08: Invite People to Group

Description: The user shall invite people to a group.

User Goals:

Dependency Use Cases:

Details:

Priority: 1

Progress Status: Incomplete

Test Phase Status: Planned

Pre-Conditions:

4.

Post-Conditions:

Trigger:

Workflow:

Requirements:

UC-SC01: Meeting Length

Description: The user shall be able to select how long they want their meeting to be.

User Goals: The user wishes to select meeting length for to be scheduled meetings.

Dependency Use Cases: Tbe user has created an account. (UC-P01)

Details:

Priority: 1

Progress Status: Incomplete

Test Phase Status: Planned

Pre-Conditions:

1. The user must have an account and the user must be in a group.

Post-Conditions:

1. The system shall ask the user if it is a one-time or recurring event.

Trigger: The user clicks create a meeting button.

Workflow:

1. User is in profile view
2. User shall select the group pane on the top panel.
3. The system shall enter the user into group view mode.
4. The user shall click the group they wish to create a meeting for.
5. The system shall display the group pane for the specific group.
6. The user shall click the create a meeting button.
7. The system shall display options for the desired length of meeting.
8. The user shall select desired meeting length.

Requirements:

1) The group leader user shall be able to set the length of the meeting that they are trying to set.
2) The system shall request verification that the meeting length is what the user intended. (make sure the user did not input typo)

UC-SC02: Recurring vs. One-Time Event

Description: The user shall be able to choose a meeting to be recurring or one-time event.

User Goals: The user wants to make a meeting

Dependency Use Cases:
Have an account. (UC-P01)
Login to the account (UC- P02)

Details:

Priority: 1

Progress Status: Incomplete

Test Phase Status: Planned

Pre-Conditions:

1. The user needs to have an account and be in a group.

Post-Conditions:

1. The system displays the available times to create such a meeting.

Trigger: The user clicks the create a meeting button.

Workflow:

1. User is in profile view.
2. User shall select the group pane on the top panel.
3. The system shall enter the user into group view mode.
4. The user shall click the group they wish to create a meeting for.
5. The system shall display the group pane for the specific group.
6. The user shall click the create a meeting button.
7. The system shall display options for the desired length of meeting.
8. The user shall select desired meeting length.
9. The system shall ask if this is a one-time or a recurring event.
10. The user shall select whether it is a recurring meeting or a one-time meeting.

Requirements:

1) The group leader user shall be able to specify whether their event is one-time or recurring.

UC-SC03: Input Personal Schedule Manually

Description: The user shall be able to manually enter his/her schedule into the web app one event at a time. The user also have the option to add a single event or a repetitive one.

User Goals: The user wants to enter his/her schedule manually instead of importing them from another source, so he/she can have complete control over the schedule.

Dependency Use Cases: Create an Account, Login to the Account

Details:

Priority: 1
Progress Status: Incomplete
Test Phase Status: Planned

Pre-Conditions:

1. The user must be logged in.
2. The user must be in his/her profile page.
3. The user shall see a weekly view of his schedule.
4. The user must have an event to enter to his/her schedule.

Post-Conditions:

1. The user must see the event in his/her weekly view of schedule.

Trigger: The user has to click on “Add Event”, enter the details of the event, and click on “Create Event”.

Workflow:

1. The user shall be in the profile view.
2. The user shall click on “Add Event” button.
3. The user shall enter the details of the event (start/end time, name of the event, location of the event, whether it is a repetitive event or a single time event, etc.).
4. The user shall click on “Create Event” button.
5. The user shall see the event on his/her weekly schedule.

Requirements:

1) The standard user shall be able to input events into their personal schedule manually.

UC-SC04: Present the Best Time

Description: The app shall find the best meeting time for a group of people. All group members must be available in suggested meeting times. If not everyone is available, the maximum number of members must be available.

User Goals: The user wants to find the best meeting time for everyone in the group to meet.

Dependency Use Cases: Create a Group

Details:

Priority: 1
Progress Status: Incomplete
Test Phase Status: Planned

Pre-Conditions:

1. The user must be a member of a group.
2. Each individual's weekly schedule must be filled.
3. All members must be invited to the group.

Post-Conditions:

1. The user shall see a weekly view of all possible meeting times where everybody can meet.
2. If not everyone can meet, alternative possible meeting times will be presented to the user.

Trigger: The user clicks on "Find the Time" button.

Workflow:

1. The user becomes a part of a group (either creates a group, or gets invited to a group)
2. After all members are added to the group, the creator of the group click on "Find the Time" button.
3. A weekly view of possible meeting time will be presented on the screen.
4. Everyone clicks on the ones that are suitable for them.
5. The meeting time with the most clicks will be chosen as the meeting time.
6. If there are more than 1 meeting time with the most clicks, all options will be presented to creator of the group in a weekly view.
7. Creator of the group clicks "choose" button to choose the desired meeting time.
8. Desired meeting time will be added to all the member's schedules.

Requirements:

1. 1The system shall get the best meeting time for the group. If not all group members are available at a given time, the maximum number of members must be available for the time slot.

UC-SC05: Be Able to View Schedule

Description: The user shall be able to view their weekly schedule.

User Goals: The user wants to view his/her schedule to be able to add/remove/update an event or just to see what he has to do for the day.

Dependency Use Cases: UC-S03

Details:

Priority: 1
Progress Status: Incomplete
Test Phase Status: Planned

Pre-Conditions:

1. User must have an account.
2. User must have the schedule input.
3. User must be in profile view.

Post-Conditions:

1. The weekly view of schedule must be presented on the screen.

Trigger: The user clicks on profile view.

Workflow:

1. The user clicks on profile view.
2. User sees the weekly schedule on the screen.

Requirements:

2. The system shall let the user view their profile.

UC-SC06: Start/End Dates for Recurring Schedule

Description: The user shall be able to choose start and end dates for their recurring events so that the algorithm doesn’t consider them after they are over.

User Goals: The user wants to specify a start and end date for repetitive events so they do not show in their schedule after they are done.

Dependency Use Cases: Create a Group, Create a Meeting, Present the Best Time

Details:

Priority: 2

Progress Status: Incomplete

Test Phase Status: Planned

Pre-Conditions:

1. The user must be in the profile view.
2. The user must be able to make an event.
3. The user must either want to add an event or want to edit an event.
4. If this is for a group, only the group leader should be able to edit this.

Post-Conditions:

1. The event is added to the schedule for a certain period of time and it is not shown before or after the start and end time.

Trigger: The user shall enter the dates manually when adding an event.

Workflow:

1. The user shall be in the profile view.
2. The user shall either want to add an event or want to edit an event.
3. The user shall enter start/end date manually when adding an event.
4. After clicking “Create Event”, the user shall see the event on his/her schedule from the start date to the end date repetitively (the event is not shown on the schedule outside of the time period).

Requirements:

1. The user shall be able to input their Start and End dates.
2. The system shall require the Start date to not be a later date then the End date.

-

UC-SC07: Re-schedule

Description: The system shall allow the user to re-compare schedules in the group while reflecting changes in each user's schedules.

User Goals: To be able to compare a new personal schedule with a group schedule and run the algorithm on them.

Dependency Use Cases: Create Profile, Create Schedule, Join a Group

Details:

Priority: 2
Progress Status: Incomplete
Test Phase Status: Planned

Pre-Conditions:

1. There are at least two schedules to compare
2. The user must be a member of a group
3. A user must have made a new personal to compare with.

Post-Conditions:

1. A diff of the group schedule and the users' schedules is returned.

Trigger: The user requested to diff the personal schedules with the group schedule.

Workflow:

1. The user shall be in the profile view.
2. The user shall be able request a difference of the group's schedule and the personal schedules of all of its members.

<u>Path 1:</u> (If there are changes to a member's personal schedule)

1. The user shall be given the re-diff of the updated schedules.

<u>Path 2:</u> (If there are no changes to a member's personal schedule)

1. The user shall be alerted that there are no changes to any member's personal schedule to diff with.

Requirements:

1. The group leader shall be able to reschedule a meeting time.
2. The system shall update a meeting time.

UC-SC08: Providing Alternative Meeting Times

Description: The system shall display multiple meeting time options to allow for edge cases in people's schedules. The system shall also display more times for fewer and fewer people for maximum options for the group.

User Goals: The user shall be able to view a list of best meeting times based on the constraint given in the group schedule compared with the members' personal schedules.

Dependency Use Cases: Created a Profile, Created a Group, Joined a Group, Created Personal Schedule, Created Group Schedule, Diff Schedules.

Details:

Priority: 1
Progress Status: Incomplete
Test Phase Status: Planned

Pre-Conditions:

1. The user shall create a new event within a group
2. The user shall edit a pre-existing event with an updated schedule list.
3. The user shall request a diff of the group schedule with the members' personal schedules.

Post-Conditions: The system shall display a list of new times dependent on the schedules of the members in the schedule list in decreasing number of available number of members. The schedule shall also display multiple times for unforeseen scheduling conflicts.

Trigger: The user requested a diff of the group schedule with the members' personal schedules.

Workflow:

1. The user shall be in the group view.
2. The user shall be able to choose constraints on meeting times.
3. The user shall request a diff of the group schedule with the members' personal schedules.
4. The user shall be given a list of best possible meeting times based on the maximum number people available during a given time block in decreasing order.

Requirements:

1. The user shall be able to view various meeting times to allow for alternative desired times.

UC-SC09: Event Names

Description: The user shall be able to name the events that they create

User Goals: User makes event distinguishable by giving it a name.

Dependency Use Cases: User shall be in a group

Details:

Priority: 1
Progress Status: Incomplete
Test Phase Status: Planned

Pre-Conditions: User is in a group, user creates event

Post-Conditions: Event becomes named

Trigger: User clicks on "Event Name" text field

Workflow:

1. User creates or edits event
2. User clicks on "Event Name" text field
3. User enters event name
4. User submits
5. System updates database
6. Event name created/changed

Requirements:

1. The standard user shall be able to name an event to distinguish it from other events.

UC-SC10: Auto-Populate Schedule from Various Sources

Description: User automatically populates his/her schedule by signing into various sources

User Goals: User's schedule includes events from multiple sources

Dependency Use Cases: User has created account

Details:

Priority: 2
Progress Status: Incomplete
Test Phase Status: Planned

Pre-Conditions:

1. User has created account

Post-Conditions:

1. User has his/her personal schedule automatically populated with events from the calendars of the accounts they have signed into

Trigger: User clicks on "Link to other accounts"

Workflow:

1. User goes to the page where one edits his/her profile
2. User is presented with the page in which he/she can edit his/her profile
3. User clicks on “Link to other accounts”
4. User is presented with different options that allow him/her to log into Facebook, Google, etc.
5. User log into one or more of the accounts.
6. System populates user’s personal schedule with the events from his/her linked accounts