Let's Meet

Ethan Christensen, Kolby Kunz, Hope Welch, Lindsay Wilde

Design Document

April 26, 2022

Table of Contents

1. Executive Summary

   Let's Meet is a scheduling app that helps people find times to meet up with ease. At one point or another, people have to schedule some sort of meeting or event with others. Oftentimes, finding an opening that works for everyone can be difficult. The purpose of this project is to remove the frustration of finding time openings. Instead, import your calendar(s) from the current scheduling or calendar app(s) you use. If you do not use one, just add your schedule to *Let's Meet* directly. Once the calendars have been created, you can then join groups with others and click a button to have the app find a time opening for your group and you to meet up.

   When coming up with the idea for this project, the idea of coordinating group projects often came to mind. When given a group project, one of the most challenging parts of it can be finding time to discuss with your fellow group mates. Instead of always having to ask someone their schedule, you can add your group mates to a *Let's Meet* group and look at the app to see everyone's availability, which already contains everyone's schedule. Another group of people that could benefit from this are families. A family could have a calendar to see what every family member is doing and when. Being able to see this information could help families better plan meals, family outings, etc.

   The main function of *Let's Meet* is a scheduling app and time coordination app with the functionality mentioned above. Another thing that we hope to add is a time management function. The purpose of this is to help users find openings in their own schedule to do homework or anything else that needs some help with managing time. Given how long you think something will take and how long you want to work on it at a given time, the app will find when it thinks is best for you to work to be able to get it done on time.

2.Background and Technical Requirements

Summary

For our project, we believe that everyone will need help with time coordination at some point. Either in their professional career, family reasons, education or even hanging out with friends. With everyone having busy schedules, we want to help people to be able to have less worries about planning event times that will work with everyone they want to see.

We want to accomplish this by using our skills of ASP.NET, constraint satisfaction problems (CSPs), React Native, AWS, Microsoft SQL server/relational databases, C#, HTML, CSS and JavaScript. As a team, we have experience with most of these things and will be able to easily pick up and learn the tools we do not know.

Overview

Let's Meet is needed because scheduling is a problem that is faced by many people. For example, those in an academic setting, business meetings or for families. When planning for future meetings or meals in a family setting, it can be difficult to find a time that works well for everyone. There is often discussion that takes place about finding these openings. With this project, the hope is to limit those discussions and focus on more important tasks at hand. When using *Let's Meet*, the app will help shorten these discussions by having everyone join a group together and share their calendars. Once the group is created and calendars are shared, the app will be able to automatically find openings for potential meetings.

The project might sound similar to something called *When is Good*. This is a website also hoping to solve the problem of scheduling meeting times in a group of people. Our project does hope to solve a similar problem of scheduling. As a group, we believe we can create something better than *When is Good.*

The main thing we can improve upon is the UX and thus the UI. *When is Good* is focused for a website or things with a large screen size. With *Let's Meet* we plan to create a mobile and a web application. The mobile app aims to be for when you're in a time crunch like in a classroom setting where you need to quickly add people to a group before class is over. With our project, there is no need to struggle with clicking a few

dozen boxes each week to state when you're available like *When is Good*. With *Let's Meet* you import your calendar from your current calendar app or input it by hand into the program. Your week is automatically filled in with your events, telling the app that you're not available to meet at those times.

Technologies

For the mobile app UI, we plan on using React Native. One reason we plan on doing this is because it will allow our app to be for both iOS and Android without having to make two separate versions for each one. As a group, we have very little or no experience with React Native. We decided that we want to learn this instead of Flutter, which was the other option that was being considered. We believe that we will be able to learn React Native. We have experience with JavaScript. With this in mind, we will be able to quickly pick up React Native. Another reason we want to use React Native is because we believe knowing how to use this will help us in industry after graduation. Since it tends to be used frequently in industry and we hope by gaining some experience with it now, it could help us with more than just capstone.

For the website UI, we plan on using .NET core. The reason being is our entire team has experience with this and C# from previous classes. Within .NET core, a web view is built from CSHTML files which use the usual HTML, CSS, and JS along with C#. Our entire team has experience with these languages.

For the web server, we plan on using AWS. Most of the team has familiarity and experience with this from the Web Software Architecture class. We decided that going with this option would be best since we are familiar with it.

We also plan on using .NET core for the backend/API. As mentioned, we all have experience with C# and have a preference towards using this language. We are using this tool because it does a lot of setup for us especially for user creation, authentication, and authorization. The ORM, Entity Framework, is also very helpful.

For the database, we are going to use Microsoft SQL server. We are using this because we are also planning on using Entity Framework core, which is another Microsoft technology. Since we are using this and lots of the .NET core framework, we have an ecosystem of technologies from the same provider. We hope that this will help us have fewer issues with connecting things. As a group, most of us have taken the database course and have experience with relational databases, which is what Microsoft SQL server is.
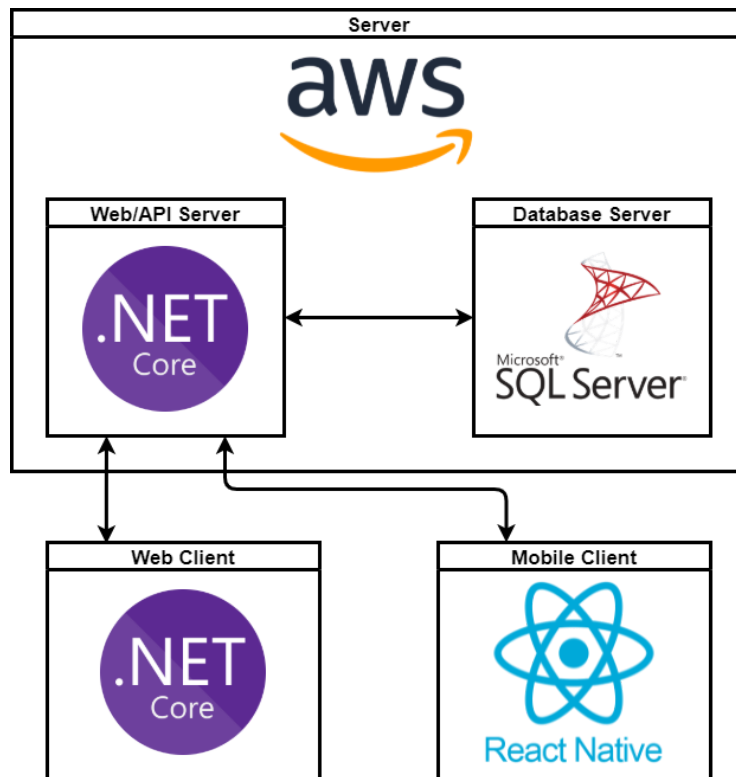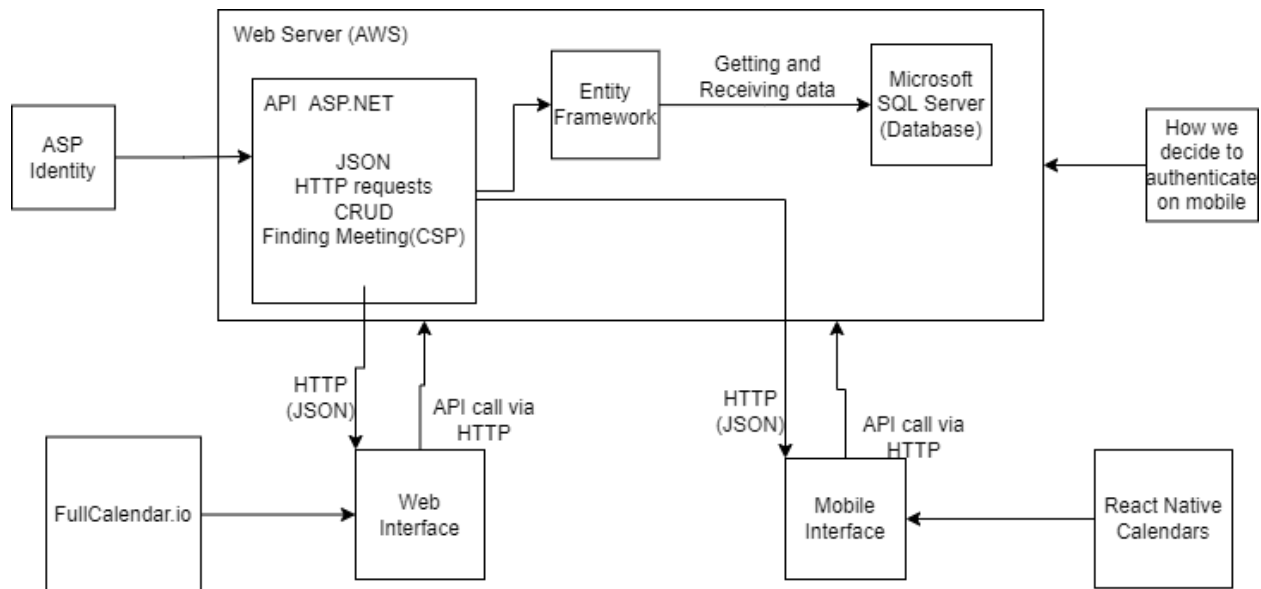
Let's Meet tech stack

Software/Hardware Requirements

With *Let's Meet,* we want to have a web page and mobile app (both iOS and Android). With this goal in mind when creating the project, the user will not need any special software or hardware. All they will need to do is download the app or visit the website.

3. Requirement Analysis

System Architecture

Web Interface - The web interface will be created in the ASP.NET cshtml pages that use HTML, CSS and JavaScript by us. More specifically, the people in charge of the frontend creation will be Hope and Lindsay. Lindsay with more focus on the web end and Hope on the mobile end.

Mobile Interface - The mobile interface will be created with React Native by us. The mobile frontend, will be in charge of by Hope and she will put most of her focus on it.

Server - We plan on hosting the server on AWS.

Database - We will be using Microsoft SQL server, a relational database. The way we are creating tables and the database is with Entity Framework.

Entity Framework - Entity Framework will allow us to treat our database as C# objects. We will have to worry a bit less about the specifics of how things are in the database. As we will also be using a "code first" approach, we will write C# objects and code to create a database through Entity Framework.

API - In order to communicate between the interfaces of web, mobile and the server, we will have API calls in the form of HTTP requests. When an interface makes an HTTP request in the form of an API call, they will receive JSON back as a response. It is then up to the interfaces to parse the JSON that was received. Within the API there is the solution to the CSP (Constraint Satisfaction Problem) of finding available time slots. This will be created by us from information taught in the AI class and other research we might do. We need to have this algorithm be efficient so users can know the result of when a group meeting can occur for everyone within seconds if not fractions of a second.

Authenticating users and login - For Authenticating users on the web site end, we plan on using ASP.NET Identity. For the mobile end, we won't be able to use Identity directly, but we still need to be able to authenticate and save those user's information. A way to do things is to work with Identity and give it the information it needs to authenticate users. One way to do this is with JWT(JSON Web Token). A way to do this is have the sign in on mobile be an API call that takes the input fields as arguments and uses that to create a mobile user in Identity. After that, the method will return a JWT or cookie depending on what react native would want as its return to know someone is logged in and authenticated. Identity will be used on both the web and mobile to authenticate the user, but at the mobile end we will have to get track of cookies or a JWT in order to make sure the user is logged in and authenticated correctly.

Calendar Libraries for mobile and web - Since a calendar is a crucial part to *Let's Meet* we went with some popular calendar libraries that we found. For the web interface, we will be using FullCalendar. This is a JavaScript based calendar. From looking at the demo on the website, it has the features we wanted a calendar library to have. Such as, different views for the calendar, event creation and time zones. This prevents us from having to recreate the wheel with our calendar. For the mobile end, we will be using React Native Calendars. This is because when looking into FullCalendar, it didn't support react native. We couldn't find an active calendar library for both mobile and web. React Native Calendars does similar things to FullCalendar, but on a mobile scale. For both of these libraries, we plan on looking into the API more and seeing how we can leverage both libraries to their full potential. Since the calendar is going to be a bit part of our frontend.

Personnel

Hope Welch
Mobile - internship experience with React (although not specifically react-native). I developed a few React pages during my internship and had to learn the language from scratch.
Web - took the Web Software Architecture class where I learned how to full stack develop a website using ASP.Net Core.
API - familiar with Postman through the Web Software Architecture class.

Lindsay Wilde
Frontend (Web mostly) - Experience with HTML, CSS and JavaScript from personal studies. If needed, I can help with the mobile end as well. I've been learning React and React Native because I've been curious about it and I hear lots of talk about it.
API - Familiar with how HTTP requests work and the language C# that we are using for our application backend.

Ethan Christensen
Database - Taken database class and worked with databases for years in industry.
Backend and Scheduling logic - Taken Web Software Architecture and Databases courses for experience in backend. Years in industry of full stack development for more backend experience. Taken AI class where we dealt with how to solve similar problems.
API - Built APIs in industry. Experience building web servers from scratch including designing API.

Kolby Kunz
Scheduling logic - Studied constraint satisfaction within the AI course of Professor Kuntz.
AWS support - Set up AWS in Web software development and set up current AWS.
Mobile - Working with React Native with current Job, mostly will be supporting info.
Backend Logic - Completed Web software architect using ASP.net.

System Features

Rank 1: Bare Essentials

B1. Ability to authenticate and login a user on the web site.
- This will be done with Identity.

B2. Ability to authenticate and login Users on the mobile end
- We probably won't be able to use Identity here, so we will have to find another way to do so.

B3. Create a personal schedule/calendar from scratch inside *Let's Meet.*

B4. Be able to join a group with other people.

B5. Automatic Meeting Scheduling - Able to take people that are in a group together and find an open meeting time that works for the entire group based on everyone's schedule.
- We plan on using CSP solving methods taught in the AI class for this.

Rank 2: Planned Features

P1. Add friends and add people to a group calendar through QR codes or links.

P2. Filtering of calendars similar to what Google Calendar does
- Where you can select what calendars you want to view on the homepage. Such as only viewing your calendar, or viewing your calendar and group A at the same time on the home page.

P3. Import an existing personal schedule from several third party apps.
- Most calendar apps use an ICS file, we will need to figure out how to get that data and how to use it.

Rank 3: Bells and Whistles

W1. Privacy toggles:
- Default in a group is no one can see your details in your schedule.
- Can choose if a group can see your details.
- No way to be added to a calendar without an emailed link or being scanned.

W2. Filtering ability within a group.
- See selected people's schedules only.
- Create an event between selected people in a group. This is creating an event that only select people on the team need to go to, but the entire team should be aware of the meeting.

W3. Time management feature
- Say you have an assignment you think will take 3 hours to do, the app will find time openings in your schedule where it thinks you could work on it.

W4. Allowing users to give priority/preference to a meeting time
- For example, say you have 30 minutes open, but you don't want to meet then unless absolutely necessary
- You would rather meet at one of your other open time slots
- In order to do this, we would need some sort of rating system. This is where users can say if they liked the suggested meeting time or not

W5. Connection issues
- Say someone on mobile has slow or no internet, how do we want to handle this problem, just save it into their local storage.

W6. Having some sort of chat system between groups and/or friends.

4.Software Engineering Tools and Techniques

Team Meetings

We plan on meeting at least once a week to touch base, see our progress for the week and decide what we should be doing in the next week. We tend to do this during class times. If there is anything else that requires everyone present, we will plan another meeting to do that. This will happen when we have something such as presentation practice or technical decisions. We will be doing this mostly over Discord or Zoom. In order to determine the person that leads the meeting, we will roll a D20 and the highest one will lead it. By leading the meeting, this person will set the agenda and ask others what they've done or what they want to talk about.

Versioning

We are planning to use Git for version control and GitHub to hold the repository. We decided to go with this since we have experience using these from other classes or personal projects.

GitHub Project Board

In order to track tasks to do, in progress, completed items and more we are using GitHub's Project Board as our Kanban board. We are using this because we wanted to minimize the things we have to keep tabs on. Since we are using GitHub for our repository, we figure using their Project Board would be useful for us.

Bug Tracking

In order to make sure everyone is aware of any bugs that currently exist in the system, there is a known *Known Bugs* column in our GitHub project board. This is where people can go to report any bugs or something weird that is happening with the project. Once someone decides to try and fix the bug and finds a fix, they can move it to

the *Fixed Bugs* column along with a short description of how they fixed the bug. This will tie into the Issues tab of the GitHub repository. We also plan on using Github's Issue board to help with bug tracking. We want to try and minimize how many tools we have to check. So continuing to use GitHub, a tool we are already using will be helpful.

Postman

In order to help us have our API be organized, we will be using *Postman*. This will help the backend personnel test the API and help them manage it easier. Along with helping the frontend personnel be able to find API calls easier. Since the frontend people might know what sort of call they want, but not where it's located in the code or what they need is called.

5. Timeline

Feature Timeline

Item	Due Dates	Responsible Person
Design Doc Update 1	9 September 2022	-Lindsay (turn in on canvas)
B1 and B2	16 September 2022	-Kolby (web/api call) -Hope (mobile)
B3	23 September 2022	-Lindsay (web) -Hope (mobile)
Alpha Demo	30 September 2022	
Design Document Update **User Guide** **B4**	21 October 2022	-Ethan (Backend) - B4 -Kolby (Design Doc updates) -Lindsay (web frontend) - B4 -Hope (mobile frontend) - B4 -Lindsay (Canvas turn ins)
Beta Demo	26 October 2022	
P1	28 October 2022	-Hope (mobile frontend) P1 -Ethan (API) P1 - Lindsay (web frontend) P1
B5	11 November 2022	-Kolby(backend) -Ethan(backend)
P2	18 November 2022	-Ethan (backend) -Lindsay(frontend web) -Hope(frontend mobile)
Final Design Doc **Final User Guide** **P3** **W1**	2 December 2022	-Lindsay (Canvas turn in) - Hope (mobile frontend) -Lindsay (web frontend) - Ethan (backend)
Demo Day!	9 December 2022	

Weekly Timeline (To help us stay organized)

Week	Description
8/22 - 8/26	Ethan: Database implementation Kolby: Update endpoint layout Hope: Updated Design Doc Lindsay: UI clean up and update current draft
8/29 - 9/2	Ethan: Database implementation Kolby: Update Identity for mobile integration Hope: Database implementation Lindsay: Implement current UI draft
9/6 - 9/9 (Labor day 9/5)	Ethan: API / Backend (CRUD) Kolby: API / Backend. Finish updating Identity. Hope: Mobile Login and Environment debug Lindsay: Implement current UI draft
9/12 - 9/16	Ethan: API / Backend Kolby: API / Backend Hope: Implement current mobile UI draft (mock/hard coded data) Lindsay: Implement current UI draft
9/19 - 9/23	Ethan: API / Backend Kolby: API / Backend Hope: API handling/displaying for mobile Lindsay: API handling/displaying for web
9/26 - 9/30	Ethan: Calendar Sync Kolby: API / Backend Hope: API handling/displaying for mobile Lindsay: API handling/displaying for web
10/3 - 10/7	Ethan: CSP Solving Kolby: CSP Solving Hope: API handling/displaying for mobile & Get people to test app if possible Lindsay: API handling/displaying for web
Fall break (10/10 - 10/14)	All: Get people to test app if possible
10/17 - 10/21	Ethan: CSP Solving Kolby: CSP Solving Hope: Revisit UI design and see if a new one would be better (color scheme and fonts) (using feedback from people) Lindsay:
10/24 - 10/28	Ethan: Misc. Backend

	Kolby: Investigate QR code capabilities Hope: Implement UI changes / implement new features Lindsay: Implement UI changes / implement new features
10/31 - 11/4	Ethan: QA Kolby: Investigate/implement QR code capabilities Hope: Implement UI changes / implement new features Lindsay: Implement UI changes / implement new features
11/7 - 11/11	Ethan: QA Kolby: Implement QR code capabilities Hope: Tie up all loose ends to prep for debugging Lindsay: Tie up all loose ends to prep for debugging
11/14 - 11/18	Ethan: Bugs & Polishing Kolby: Bugs & Polishing Hope: Debugging / polish Lindsay: Debugging, polish and Documentation
11/21 - 11/23	Ethan: Bugs & Polishing Kolby: Bugs & Polishing Hope: Debugging / polish Lindsay: Debugging, polish and Documentation
11/28 - 12/2	Ethan: Bugs & Polishing Kolby: Bugs & Polishing Hope: Debugging / polish Lindsay: Debugging, polish and Documentation
12/5 - 12/8 (Reading day 12/9)	Ethan: Finalizing & Presentation Prep Kolby: Finalizing & Publishing Hope: Finalizing & Presentation Prep & Publishing Lindsay: Finalizing & Presentation Prep & Documentation Polishing

6. Appendix A: UI Sketches

Web Version:

WV1 - Login page and registration flow (including import calendar from external party)

Let's Meet   Sign Up   Log In

Welcome to *Let's Meet*

Sign Up   [Log In]

Schedule Meetings
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Form Teams
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Time Manage
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Let's Meet

Log In
Email
[Email]
Password
[Password]

Log In   Don't have an account? Sign Up

Sign Up

First Name

First Name

Last Name

Last Name

Email

Email

Confirm Email

Confirm Email

Password

Password

Confirm Password

Confirm Password
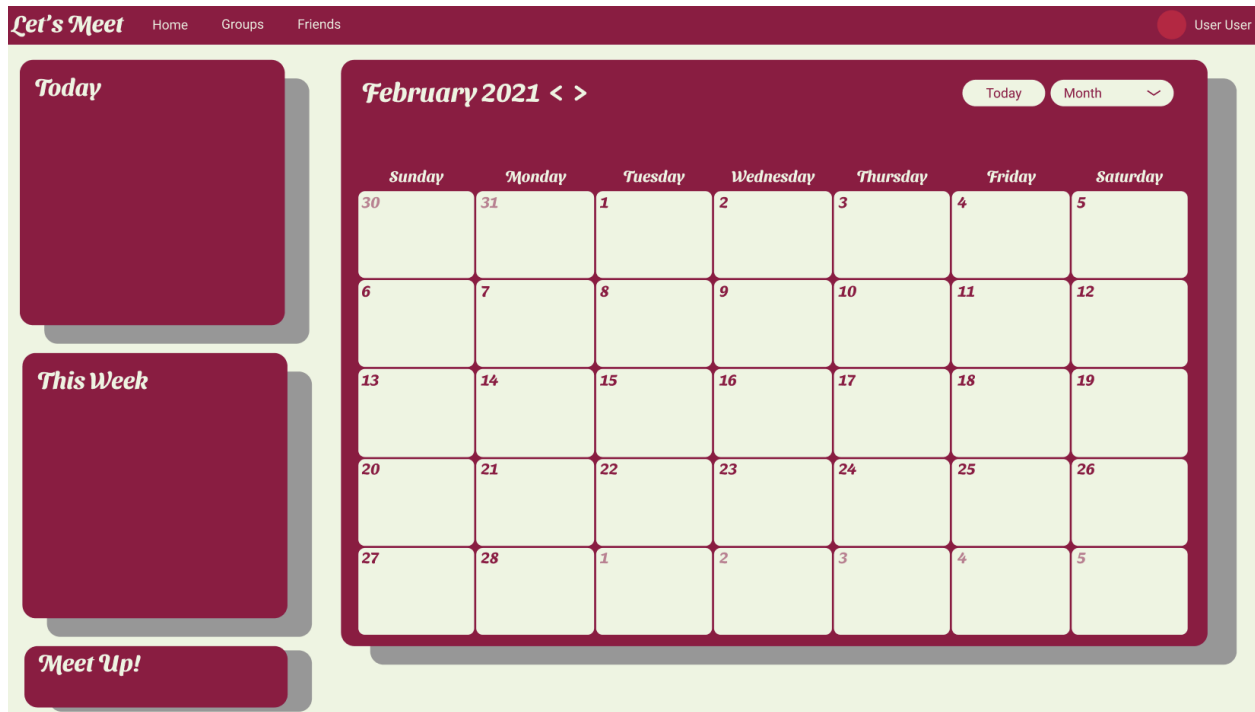
Sign Up   Already have an account? Log In

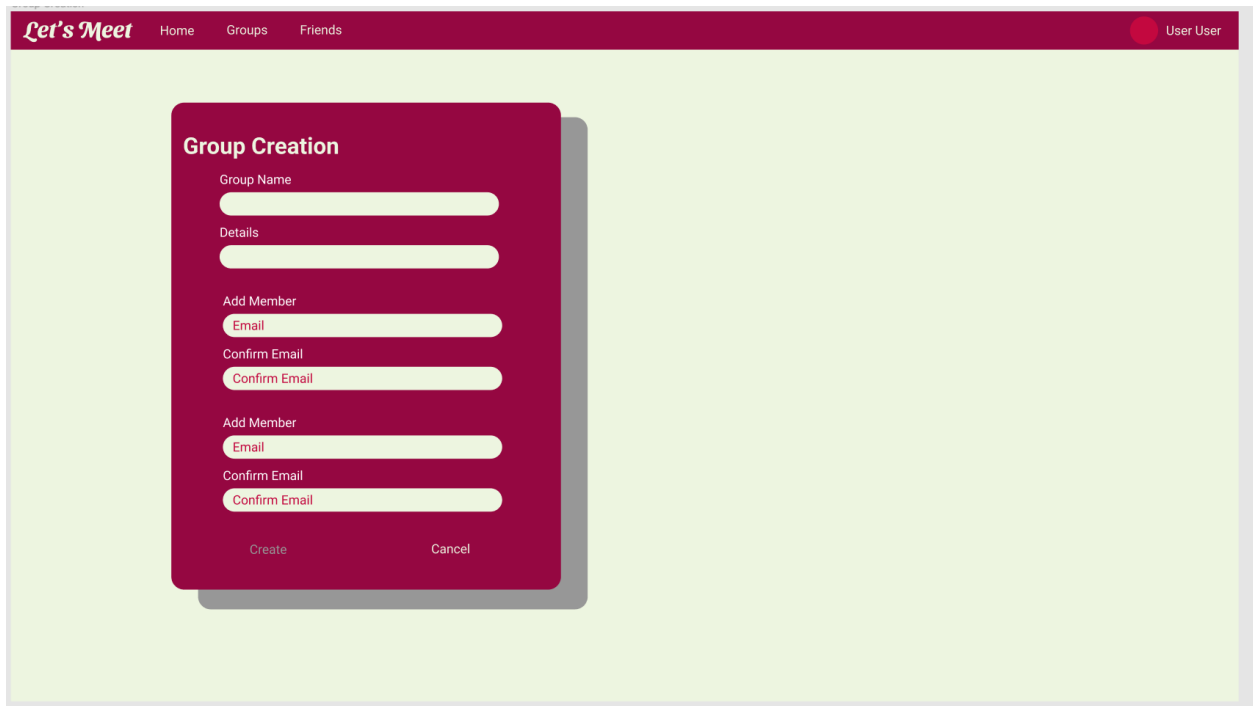**Please Verify Your Email Address**

WV2 - Main page (personal calendar)

Let's Meet Home Groups Friends User User

Today

This Week

Meet Up!

February 2021 < > Today Month ⌄

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	1	2	3	4	5

WV3 - Navigation bar (profile, personal calendar, groups, settings, Notifications)

Let's Meet Home Groups Friends User User

WV4 - Profile (user information, create calendar from scratch)

Let's Meet Home Groups Friends User User

General Privacy Plugins Account

WV5 - Create Group flow (add group, QR link to copy)

Let's Meet   Home   Groups   Friends   User User

Create Group

Let's Meet   Home   Groups   Friends   User User

Group Creation

Group Name

Details

Add Member
Email

Confirm Email
Confirm Email

Add Member
Email

Confirm Email
Confirm Email

Create   Cancel
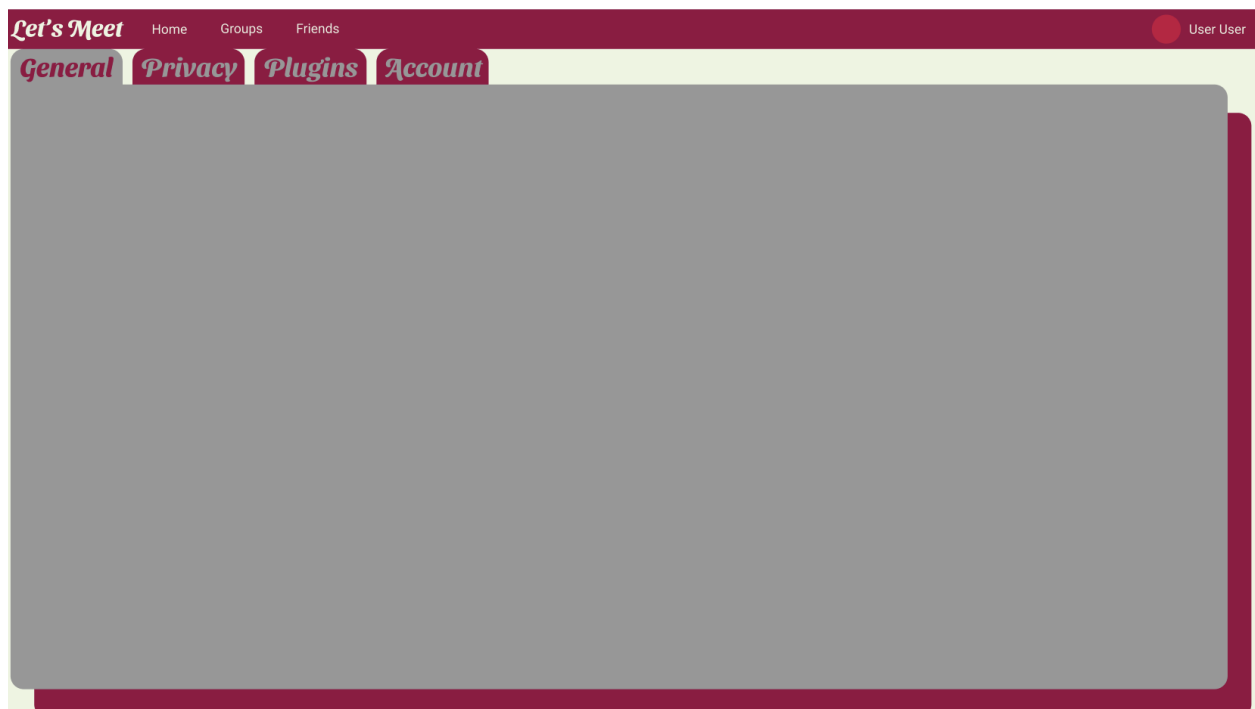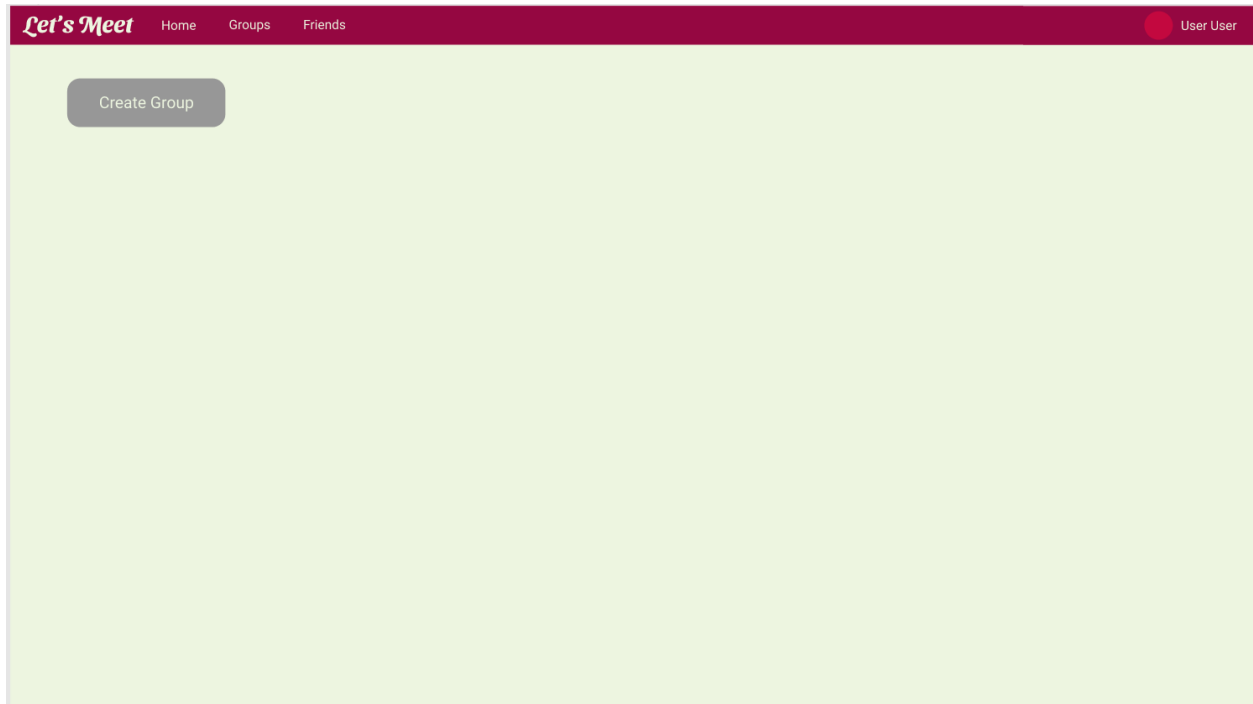
Mobile Version:

MV1 - Login page and registration flow (including import calendar from external party)

LOADING PAGE

Let's Meet

LOG IN/SIGN UP PAGE

Let's Meet

No account? Click to sign up

Username

Password

Go!

SIGN UP PAGE

Sign up for

Let's Meet

Username

Password

Email

Create Account

User Set Up Step 3.1 Class Card

What app do you use?

Sloocorp.com

Wanna manually enter in your schedule?

Yes   No

User Set Up Step 1.1

Step 1:

Do you use a scheduling app already?

Yes

No

User Set Up Step 1.2

Step 1:

Let's import your schedule!

Click on the schedule service you use from this list:

Google Calendar
iCalendar
iStudiez Pro
Canvas
. . . .

Can't find your app?

User Set Up Step 1.3

Step 1:

Hang on while we grab your schedule!

DONE

User Set Up Step 2.1

Step 2:

Are you in a rush to schedule a meeting?

Yes

No

User Set Up Step 2.2

Step 2:

Cool! We'll give you the tour next time :)
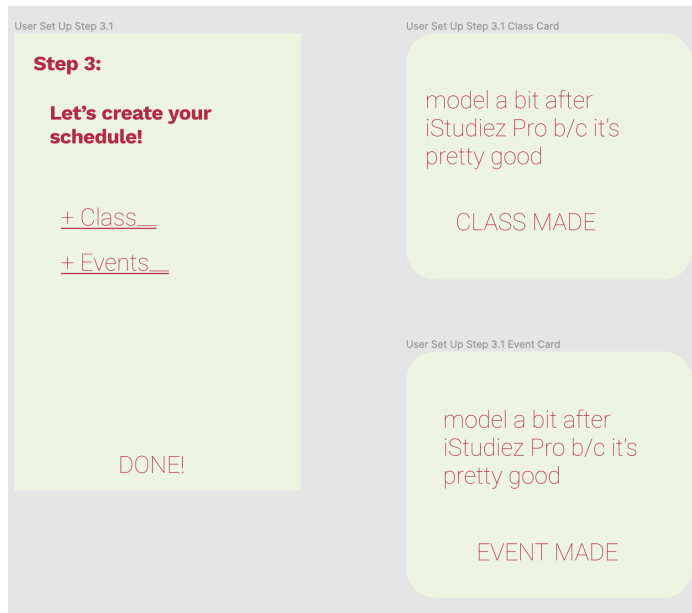
Pull up QR Code . . .

User Set Up Step 2.3

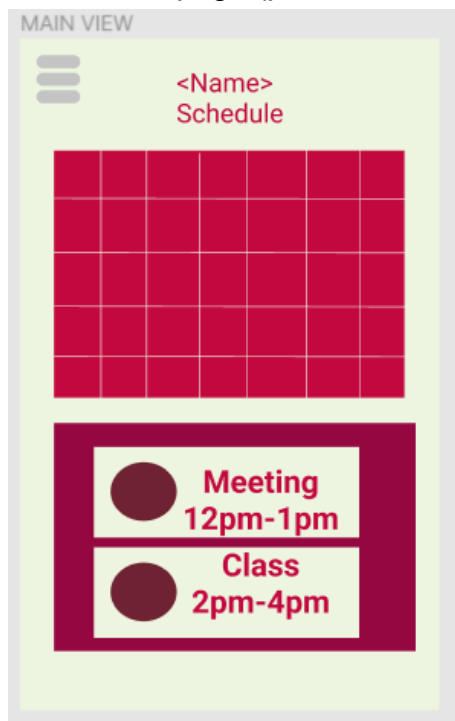**Step 2:**

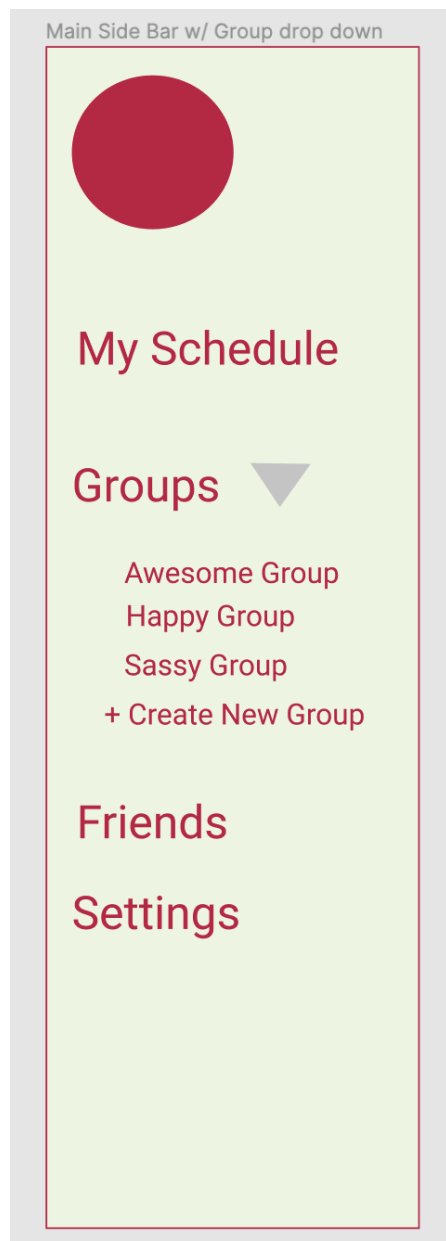**Great! Let's set up some stuff . . .**

First Name

Last Name

Next >

22

Step 3:

Let's create your schedule!

+ Class__

+ Events__

DONE!

model a bit after iStudiez Pro b/c it's pretty good

CLASS MADE

model a bit after iStudiez Pro b/c it's pretty good

EVENT MADE

MV2 - Main page (personal calendar)

MAIN VIEW

<Name> Schedule

Meeting
12pm-1pm

Class
2pm-4pm

MV3 - Navigation bar (profile, personal calendar, groups, settings)

Main Side Bar w/ Group drop down

My Schedule

Groups ▼

Awesome Group
Happy Group
Sassy Group
+ Create New Group

Friends

Settings

MV4 - Profile (user information, create calendar from scratch)

Settings

Change Profile picture

Change
password

MV5 -Create Group flow (add group, QR link to copy)

Groups

Create New
Group

Go to group

Go to group

Go to group

7. Appendix B: Use Cases

Number	Use Case 1
Name	Register a new user.
Description	The user will be required to have an account in order to use the application. When registering, they can also choose to sync their current calendar app or not.
Preconditions	User has to launch the application or visit the website.

List of Steps	1. Open the app or visit the website.
	2. Click on register new account.
	3. Input first name, last name, email, confirm email, password and confirm password.
	    a. Will be asked to confirm the email.
	4. The user will then hit next and be given another page asking the question "Would you like to sync a calendar from another application?" If the user selects yes, go to step 5. If the user selects no, go to step 6.
	    a. The option to sync their calendar or not can be changed in their profile at any time when using the app.
	5. The user selected yes to calendar syncing. The user will be asked what calendar system they currently use and select it and give permissions to *Let's Meet*. Then that calendar will be synced with their *Let's Meet* calendar.
	6. The user selected no to calendar syncing. Nothing else needs to happen.

Related UI	WV1 and MV1

Number	Use Case 2
Name	User wants to add a friend.
Description	Users will be able to have a friends list so they can easily add people they know already to groups.
Preconditions	Use Case 1
List of Steps	1. The user will log into the app or website. 2. Navigate to the "Friends" tab in the menu and click to open the friends list.

	3. Once in the friends list, there is an option to add a friend. 4. Click on the add friend icon and a popup appears and the user will now be prompted to enter the friend's email address. 5. Once the information is inputted, the user will have to wait for their friend request to be accepted or declined. a. See use Case 9 for how it will be accepted or declined.
Related UI	WV3 and MV3

Number	Use Case 3
Name	User wants to create a group without QR code.
Description	Users must be in a group in order to find meeting times between each other and coordinate time within a group, so creation of a group is a must for use case 4.
Preconditions	Use case 1 and optionally use case 2
List of Steps	1. The user will navigate to the "Groups" tab in the menu. 2. The user will now see a list of the groups they are in and the option to create a new group with QR code or from searching friends list/inputting email addresses. 3. The user will select "Create a new group from email", something will pop up asking for the group name and privacy settings for the group. Along with asking if there are any people you would like to add to the group now 4. In the group creation menu, select add members. 5. When adding members without a QR code, you will input the user's email address or add from friends list and they will have to accept the group invite. Once accepted, the users are in a group together. a. See use case 11 for accepting group invites.
Related UI	WV5 and MV5

Number	Use Case 4
Name	User wants to create a meeting between group members.
Description	This is the part that will use CSP(Constraint Satisfaction Problem) to quickly find an opening between group members. This is also the "Let's Meet" part of our app, time coordination.

Preconditions	Use case 3 or Use case 5

List of Steps	1. The user will navigate and open the "Groups" tab in the menu.
	2. The user will then navigate to the desired group in the list of groups they are in and select the desired group.
	3. Once on the group page, they will see the calendar for the selected group.
	    a. On the group calendar, it will show times when group mates are busy.
	    b. It will also show other meetings that the group might already be scheduled for.
	    c. Both of the above are considered blocked out time that a meeting cannot take place at.
	4. At the top of the group calendar, there will be a button labeled "Meet up". When you click on this it will open a menu with meeting options.
	    a. It will ask between what days you would like the meeting to take place. It will also ask how long you intend the meeting to take.
	5. After filling in the information about the meeting and selecting "Create Event", group members will receive a notification about the meeting.
	    a. For information on accepting or declining a meeting see use case 8

Related UI	WV2 and MV1

Number	Use Case 5
Name	User wants to create a group with QR code. (This will be a mobile app specific)
Description	The user uses a QR code to make it simpler when in a "on the go" situation, like being in class. No need to accept or decline the group invite, with the QR code scanning, users will automatically be added to the group.
Preconditions	Similar to use case 3, but this use case 3 is not required, no need to input emails, instead scan QR codes.
List of Steps	1. Open the mobile app and navigate to the "Groups" tab. 2. Select "Create Group" and then the "Create QR code Group" option. 3. The one creating the group has the group QR code on their phone and the other group members will scan this to join the

group.
4. In order to join the group, users will go to their profile, which is located on the menu bar.
5. In their profile, they will go to the option "Scan QR code", this will open the camera and all they have to do is simply scan the QR code of the group from the user that selected "Create QR code group".
    a. With using the QR code group creation, there is no need to accept/decline a group invite like in use case 3.
6. After scanning the QR code, the user will see a pop up asking to confirm they want to join the group.
7. Once everyone has scanned the QR code and joined the group, the user with the QR code open can select the option at the bottom that says "That's everyone" to let the app know that everyone has joined the group for the time being.

Related UI	

Number	Use Case 6
Name	A user wants to sync their calendar from a different calendar app with *Let's Meet* after registering their account.
Description	If a user has a calendar located elsewhere, users will have the option to sync that calendar into the app. The calendars between the two applications should stay synced together.
Preconditions	Use case 1
List of Steps	1. User will navigate to their Profile in the menu 2. User will select the "Sync Calendars" option 3. In the "Sync Calendars" menu, the user will select what other calendar they use from a list of options. 4. After selecting the other calendar they use, they will have to give permission to *Let's Meet* and then things will be synced.
Related UI	MV1

Number	Use Case 7
Name	Users need a way to edit their individual calendar within the app.
Description	If a user doesn't use another calendar app or doesn't want to sync,

	they have the option to create and edit their calendar within the app. Along with the option to edit their already existing calendar they imported within the app. Any changes made in *Let's Meet* should sync to their other calendar.
Preconditions	Use case 1

List of Steps	1. The user will navigate to the "Home" tab on the menu bar.
	    a. The "Home" tab displays a calendar that has all their events going on. Including ones from groups and individual ones.
	2. Near the calendar, there will be a "Create event" button, after a user clicks on that to create an event, a small window opens.
	    a. Events created with the "Create event" button are assumed to be for an individual only. Since for a group event, a notification goes out to the team before the event is confirmed.
	3. The window that opens will ask for details about the event.
	    a. Required details include date, start time, endtime
	    b. Optional details include, name, details, location
	4. Click to confirm event creation after inputting the needed information, the event will now show up on your calendar.
	Alternative way
	1. The user will navigate to the "Home" tab on the menu bar
	    a. The "Home" tab displays a calendar that has all their events going on. Including ones from groups and individual ones.
	2. The user will click on a day in the calendar and the system will bring up a window to input information about the event.
	3. This time, the day will be filled in automatically and the user will have input the start and end time of the event.
	    a. Along with optional details including, event name, details, location.
	4. Click to confirm the event after inputting the required information and now the event shows up in the user's calendar.

Related UI	WV1, WV4 and MV1

Number	Use Case 8
Name	Accepting or declining a meeting time.

Description	A groupmate has asked to meet up, the other members need a way to say that meeting time works for them or not. Even if the algorithm said they were open during a time, we want group members to be able to confirm the time works.
Preconditions	Use Case 4

List of Steps	1. The user will navigate to "Notifications" in the menu bar
	2. Upon clicking on "Notifications", the user will see something like "User C from Group Q has asked to meet up on April 12 at 12pm, would that work for you?" The user would then click accept or decline
	    a. If the user accepts and all other groupmates do, the meeting will appear on the group calendar and everyone's individual calendar
	    b. If a single groupmate declines the meeting, the user that asked for the meeting will be notified that the meeting could not occur because it didn't work for X group mate.
	       i. When the meeting is declined, the user that asked for the meeting will be notified why and they will need to create a new event and ask the group again.

Related UI	WV3

Number	Use Case 9
Name	Accepting or declining friend request
Description	A user needs to agree to being friends with someone before they are added to the friends list.
Preconditions	Use case 2
List of Steps	1. The user will navigate to "Notifications" on the menu bar. 2. In the "Notifications" drop down, it will show group invites, friend requests and any other notifications related to the app. 3. The user can select the friend request notification and click accept or decline. Alternative way 1. The user will navigate to "Friends" on the menu bar 2. In the "Friends" window the user will see their list of friends. Above that will be a notification for a pending friend request. 3. When the user sees the pending invite at the top, they can

	select accept or decline.
Related UI	

Number	Use Case 10
Name	The user wants to view their calendars.
Description	Users will be able to view their individual calendar, a group calendar, or multiple group calendars. A User should be able to filter between what calendars they want shown on their home page.
Preconditions	Use case 1 or Use case 3
List of Steps	1. The user will navigate and click on the "Home" tab in the menu bar. 2. Once the user has clicked on "Home", the calendar will be displayed, this calendar displays things based on the user's filter 3. If user would like to filter the calendar to show a group calendar, multiple group calendars or the individual user calendar, there is an option to do so a. On the side of the calendar there is a filtering option. Users will be able to check or uncheck what groups they want displayed.
Related UI	WV2 and MV2

Number	Use Case 11
Name	Accepting or declining a group invite.
Description	The users will need a way to agree to joining a group.
Preconditions	Use case 3
List of Steps	1. The user will navigate to "Notifications" on the menu bar. 2. In the "Notifications" drop down, it will show group invites, friend requests and any other notifications related to the app. 3. The user can select the group invite notification and click accept or decline. Alternate way 1. The user will navigate to "Groups" on the menu bar

	2. In the "Groups" window the user will see the list of groups they are in. Above that will be a notification for a pending group invite. 3. When the user sees the pending invite at the top, they can select accept or decline.
Related UI	WV3