CS673 Software Engineering
Team 3 : PlanningJam
Meeting Minutes

All meeting minutes are kept in this single document. The latest meeting minutes should be at the beginning of the document. For example, meeting 3 minutes is placed before meeting 2 in the document. The team leader should prepare a basic agenda for the meeting and team members should rotate to be the minutes taker. Each group should have at least one meeting per week, and you may have multiple meetings if needed.

Meeting 5

Date and Time: 7:15 PM Eastern 10/3
Place: Zoom
Participants: Ashley, David, Donjay, Haolin, Jason
Minutes taker: Donjay Barit
Timekeeper: Donjay Barit (7:15pm - 7:30PM)
Purpose: Sync on iteration 2

Agenda:

Said during lecture:
Should have most of the features implemented by now… would have been nice to know this during iteration 0!
Front end:
Edit/Make Plan
Add friends, remove friends, accept friend requests

Discussions:
- Review Git Pull Requests
- Most features done by iteration 2
    - Do what we can to get most of them done by iteration 2 submission
    - Connect frontend with backend
- Debugging and Testing iteration 3
- Iteration 2 Presentation
    - Ashley and Jason will present
    - Ashley will start on presentation slides
    - Presentation recording on Monday night
- Outstanding Work:
    - Ashley waiting for friends and plans filters

    - Jason - Friends API documentation
    - Haolin - RSVP fix current pull request, User info API
    - Donjay - Plans/Friends filter
    - David - Frontend Friends UI
- Documentations
    - Continue to document API and Tests

Key Decisions:
Backend - Finish up the friends, plans, rsvp api
Create API documentations for frontend.
Frontend - Get UI for friends, plans and connect with api

Action Items:
Everyone, by Monday afternoon:
Two weeks of record-keeping
AI log
AI-human balance at start of files
Fill in aspects of STD that you can

Presentation people:
Do presentation
David:
Fill out Risk assessment with moving situation and marathon situation
Summarize progress report
Edit SPPP's timeline
Upload everything
Make release ONCE DONE

Meeting 4

Date and Time: 8:00-8:12 PM Eastern 9/28
Place: Zoom
Participants: Ashley, David, Donjay, Haolin, Jason
Minutes taker: Ashley Sachdeva
Timekeeper: Ashley Sachdeva - 12 minutes
Purpose: Sync on iteration 2 and lab

Agenda:
Lab 3: Everyone is setup to create and build tests
Who is doing what?

David and Ashley:
David takes on friends request pages and add friends
Ashley takes on Post a plan get own user plans

Backend team has apis divided amongst them for post plans, add friend
Jason and Ashley will do presentation for iteration 2
Team works on STD document now that tests are solidified for both frontend and backend.
Quiz 2
Project Iteration 2 due Tuesday, October 7 at 6:00 AM:
Haolin to sync on issue with environment variable in the discord channel

Essential:

- Friend Requests
    - Backend
        - When two people create a friend request, they are then friends
        - Accepting a friend request is the same as making one
    - Friends page
        - How to add friend
            - Search/code
        - List of friends
            - Remove
        - List of friend requests
            - Dismiss
            - accept
- Have people be able to create plans
    - Backend
        - Table
            - Title
            - Description
            - Time
            - Place
    - Frontend
        - Form

- Filter plans by type
    - List of plans by friend
    - (Don't do this yes)
    - Just use the material table features

Discussions:

Key Decisions:

Action Items:

Meeting 3

Date and Time: 5:00PM - 5:30 PM Eastern, 9/18
Place: Zoom
Participants: Ashley, David, Donjay, Haolin, Jason
Minutes taker: Haolin Yang
Timekeeper:
Purpose:
Discuss SDD and deliverable codes.
Agenda:
Iteration 1:

- What's done so far
  - Setup
  - Backend
  - Frontend
- What needs to be done
  - ProjectEvaluationRubric
  - Working code
    - At the start of each file, comment which percentage is generated by the framework, how much is generated by the AI tool, how much is human written. Also how much is altered other people's human code? Might send out more details later. It was very confusing.
    - Docker container, how to run, OR hosted. Unclear if our docker system is wanted, because she doesn't want packages?
    - 1 or 2 implemented features! Make sure it actually works
  - SDD (4:40 9/16 lecture)
    - It will be updated in the future
    - Most important is the software architecture section
      - While we are doing agile, we should still decide on architecture

            - Diagrams and descriptive text
            - Why you chose this
        - Some high level class diagram
        - Initial design of front end. Screenshots
        - Database design
        - Security: Module 6
        - Business Logic/Key algorithms:
            - Don’t put trivial algorithms
            - AI/machine learning modules
        - Design patterns: during this module, but also ones outside the module
        -
    - SPPP
        - Timeline changes
        - Address comments
    - Jira
        - User Stories: are they properly described, estimated?, and managed
    - Testing (low weight)
    - Maybe start STD
    - Progress report

Due:

- Project Iteration 1 due Tuesday, September 23 at 6:00 AM ET
- Project Midterm Self and Peer Review due Thursday, September 25 at 6:00 AM ET

Discussions:

- Project Development Updates and Task Assignments
- Discussion on Software Architecture and Security Integration
- Design Patterns and API Documentation Discussion
- Project Updates and Presentation Planning
- Database Design and Software Testing Discussion

Key Decisions:

- The group will focus on completing the SDD as the main priority for this week, with code deliverables (such as registration and login features) as secondary priorities.
- JWT tokens for authentication should expire after one hour, and the frontend will

check token validity every 10 minutes to determine if users should be redirected to the login page.
- For the friends feature, “add a friend” will take priority over the full friends list, but only after the SDD is completed.
- Use the MVC (Model-View-Controller) architecture, with React for the frontend and Django for the backend, and will document their rationale for this choice.
- For security, the team will set up GitHub's own AI security features instead of SonarQube, as permissions for SonarQube are not available.
- Each member was assigned sections of the SDD and related documents, such as software architecture, business logic, design patterns, and database design. Members agreed to share diagrams and provide feedback to each other.
- Update their progress reports weekly and ensure all required documentation (including API documentation and risk management) is up to date.
- Presentation responsibilities were assigned, with Haolin agreeing to help as design implementation lead for this iteration, and Ashley taking the next iteration.
- The team agreed to document manual testing steps and link them to tickets as comments.
- For the friends feature backend API, Jason will proceed with its creation, and Donjay will finish setting up CI/CD.

Action Items:

- Ashley Sachdeva will push her branch to the repository for David Metraux to pull.
- Haolin Yang will draft a class diagram to identify the model-view-controller interactions.
- David Metraux will work on the software architecture section of the project.
- Donjay Barit will delete any redundant user stories that are not needed.
- Donjay Barit will finish the CI/CD part of the project and get it running.
- Haolin Yang will ensure to share the class diagram with the team for merging.
- David Metraux will ask about the testing format required for the project.
- David Metraux will work on the front-end page and header for the project.

Meeting 2

Date and Time: 11:00-11:40 AM Eastern, 9/14
Place: Zoom
Participants: Ashley, David, Donjay, Haolin, Jason
Minutes taker: Jason

Timekeeper: David
Purpose:
Discuss Lab 2 and Iteration 1 Requirements

Agenda:
Lab 2 due Tuesday, Starting SDD

Project Iteration 1 due Tuesday, September 23, 2025

- Essential:
    - Application (front end and backend) up
    - User Authentication
    - Friend Requests
- Desirable:
    - User Profile Management

Discussions:

- Discuss/decide which User Stories each of us will use from the SPPP
- After the Lab has been graded, we can decide which user stories to keep/want or need
- Clarifying for Lab 2, between the 2 user stories we create, we just create a relationship diagram between them
- Look into SDD/Plan a Doodle for next Meeting in regards to the Iteration 1
- Donjay lifesaver for setting up all the environments and CI/CD
- Use docker-compose.yml in the Lab
- Docker file exists in the backend and frontend, if you want to run them separately, you can use Docker Run
- Docker compose in the Root Directory can be run at the same time
- Staging and Production credentials exist in the login/credentials document
- Took a look at SDD for Iteration 1 next week
- One person might be doing the/combining all the class diagrams after we all do them in Lab2. But the class diagrams might not be too complex. We can maybe just merge them all.
- For Iteration 1, it seems like to have to have Frontend and Backend up, and make it so that we can Login and Friend Request (alternative is “Post a Plan”, but makes sense to do Friend Request first)
- Simple Friend Request option: Have unique usernames and you can search up a username. Or we can list every user
- Order of Operations regarding

- Can use Firebase email sign in with Google/Facebook/etc for Login in the backend, but will need to let Frontend know
- For now, user authentication stuff, we can start with email & password and then we can see what Frontend needs for user profiles, will have to communicate with what else they need other than name, email password
- Documentations are kind of confusing, so we can take a look at previous classes and see how they did it
-

Key Decisions:
- For Lab2, we can highlight the user stories we want from the SPPP
- Simple Friend Request Feature first (consider Post a Plan later)
- For backend, we'll start with email and password and then see later what frontend needs for user profiles

Action Items:
- Mark/Highlight the user stories you want in the SPPP
- Merge-in PJ-21
- Can start building out UI, using dummy data until the endpoints are ready
- David: Friend Page
- Ashley: Login/sign-up
- Donjay, Haolin, & Jason will split up the back-end. Need to create some more tickets first
- Frontend/Backend communicate on what data is needed (Option could be utilizing Firebase for sign-ins)
- Link Jira Tasks for backend - frontend data communication
- Everything is in Development Branch, ready for Lab 2
- Fill out Time Progress Reports
- Haolin will take a crack at the user authentication endpoint
- Jason will look at JWT
- Donjay PR for MongoDB backend, and will help create tasks in backend, and we'll choose more later
- Setup the Friends section in Database (setup user data, user authentication, and friends data)
- Setup Unit Testing for Database

Jira Meeting

Date and Time: 4:30 Eastern, 9/11
Place: Zoom
Participants: David, Donjay, Haolin, Jason
Minutes taker: David
Timekeeper: None
Purpose:
Agenda:
Learn about Jira

Discussions:
Added Icebox that's part of lab 2: What's on hold
Backlog:
Iteration 2+3 will be in backlog while we work on iteration 1
We can choose which one to work on
Can't connect Git and Jira! Ooops
Can't do API stuff until backend environment is up
We have to start from Backlog when we want to create a ticket
Can create subtasks
Can say certain tasks are blocked by other tasks, or relates to another task.
Kanban-> Timeline -> Epic will help us for lab 2. Description of a feature. Each of these would have the subtasks for the features.
In Scrum screen, can drag stuff from the backlog to various iterations.
Scrum's Active Sprints will show Kanban Open Tasks
Good to write comments for people doing code review
Want to make sure to, when working on stuff, whether it relates to the front and back end branch?
Once it's in production, we'll QA it, and once it's in release, we'll create a version, and can start linking up all the tasks part of the release. Which would be independent from Git's release.

Jira Workflow Notes is always open

Backend will mostly just be API stuff

We can

Key Decisions:
Start putting up servers ASAP
If we can:
Create tasks
-create subtasks

Action Items:
Start putting up servers ASAP
If people can:
Create tasks
-create subtasks

Meeting 1

Date and Time: 2:30 Eastern, 9/7
Place: Zoom
Participants: Ashley, David, Donjay, Haolin, Jason
Minutes taker: David
Timekeeper: None
Purpose: Initial plans
Agenda:
Discuss which project we're going to do

- Haolin's idea
- Ashley's idea

Come up with project name
Roles
By Tuesday Evening:

- Update team.md
- We merge

What we need to do by Wednesday Evening:

- **CS673_SPPP**
- **CS673_SPPP_RiskManagement**
- Fill out individual progress reports
- **CS673_ProgressReport_team3**
- Do presentation video/powerpoint?
- Upload all of these to git

Discussions:
Started using AI companion
team.md was made for lab
Discuss roles and responsibilities: wait until we see the projects
Ashley’s idea: place to go where you create a plan. Lets people respond to it.
Like partiful, but without
Ours is nice because there is only one way to respond. Calendar view, user preferences, etc.
Haolin’s idea: AI web application
Like websim.com.
Has a basic prototype with images.
vibecoding. Just generates websites.
Example lets us play games
Code the website, the backend agent, handle the interactions with files.
Like a github that you can play.
David thinks it might be hard to do Haolin’s idea. Ashley suggests it might make sense to use some AI in the schedule planners.
Donjay says we can use machine learning to see what users would likely attend.
Jason also agrees.
Donjay says we can swap around: make sure to ask her.
Decide what we want to use:
Django.
Donjay has used hosting stuff before! Yay!
Render.com, netlify.com, fly.io, ? Etc.
Integrates with CI/CD to push the projects into those servers.
Use Jest for testing? Sonarcube for analysis.
Donjay mentions Render lets us use their own database, sql/mongo.
Haolin says we can use Django and React. Can use material library.
We’ll have a server for the front and backend, use API to connect those two.
Repo branch for backend, one for front end, then main will have a folder for both.
Use iteration 1 branch, then merge into main.
Use Jira for issues. Can test in Jira.
Can use AI for name. WE GOT A LIST.
PlanJAM. -AIgenned
Purple jam logo?
We’ll get lab 1 done.
Start discussing who is doing what in iteration 0, what is needed.

Key Decisions:
So far:
We’re doing schedule planner called PlanningJAM
We’re going to use Django and React.
We’re going to use Jira for issues
We’re going to have each iteration be a branch before merging to main.

General Roles:
Ashley: Requirement Leader
David: Team Leader
Donjay: Configuration
Jason: QA
Haolin: Design and Implementation Leader

Decided on who is doing what for iteration 0.

Action Items:

Donjay- make Jira board
David
Make recording on Wednesday
Merge lab 1 on Tuesday
On Wednesday: make iteration 0 branch, put all the stuff in it, merge it into main, make release.

Ashley- ReadMe file description of project
Have people become familiar with Jira, Django, and React.

- Readme file - Ashley
- About / description - Ashley
- Description about project - Ashley
- And release for iteration 0 submission
- Meeting Minute - David
- Progress Report - Individual
- SPPP
    - Overview - Ashley

        - Related Work - Haolin
            - https://partiful.com/
        - Proposed high level requirements - Ashley
        - Management Plan - David
        - Configuration Management plan - Donjay
        - Quality management plan - Jason
        - Ai Usage - Individual
        - References - Individual
        - Glossary - Individual
- Demo - David

  Submission on blackboard/merges - David