Project Overview

Week 2

Success is not final. Failure is not fatal. It is the courage to continue that counts.

— Winston Churchill

Project Goal

- Experience the end-to-end process of developing a software product as a team
- Build a useful and innovative mobile AI application system

Project Teams

- Teams will be self-formed, with 5 members per team.
- One TA will be assigned to each team
  - You will be soon added to a channel that includes all the teams assigned to your TA.

Project Scope: Topic

- The project topic and scope are free of choice as long as you build a mobile AI software
- Remember a few things:
  - Usefulness: The app should be useful to a specific user group
  - Innovation: The app should be different from existing solutions
  - Technical Strength: The app should have at least one strong technical component
  - Feasibility: The implementation should be feasible within the semester by the term

Project Scope: Frontend

- Android-based mobile software is recommended
  - We provide up to 2 android mobile devices per team
  - We have Android tutorial sessions
  - You can try other platforms if all your team members agree and are confident to self-learn
- Some teams can build software for other type of mobile devices (e.g., watches, AR/VR devices, etc.)
  - Discuss this with the TA if your team is interested

Project Scope: Backend and Libraries

- The Django backend is recommended as we convers this in tutorial.

- Backend of your project should be deployed on a cloud (e.g., Amazon EC2, Microsoft Azure)
  - Free tiers should be sufficient for the course project
  - Contact the TAs otherwise

- Feel free to use frameworks or external libraries. Make sure to discuss with your team members

Project Milestones (1/3)

- Initial Project Proposal
  - Submission: 2-page proposal of your project
  - Deadline: **Sun. 09/14 23:59**
- Refined Project Proposal
  - Submission: refined version of your project proposal
  - Deadline: **Sun. 09/21 23:59**
- Midterm Presentation
  - Submission: midterm presentation slides
  - Deadline: **Sun. 10/26 23:59**
  - Presentations: 10/28, 10/30 classes

Project Milestones (2/3)

- Heuristic Evaluation (HE)
  - Submission: the app apk (with major features and UI elements)
  - Deadline: **Sun. 11/02 23:59**
  - Heuristic Evaluation: 11/06
- User Acceptance Test (UAT)
  - Submission: the final app apk (with all promised features)
  - Deadline: **Sun. 11/30 23:59**
  - UAT: 12/02, 12/04 classes

Project Milestones (3/3)

- Project Deadline
  - Submission: everything
  - Deadline: **Sun. 12/7 23:59**. Freeze git repo after this
  - Final Presentations: 12/9, 12/11 classes

Iteration-Wise Checkup

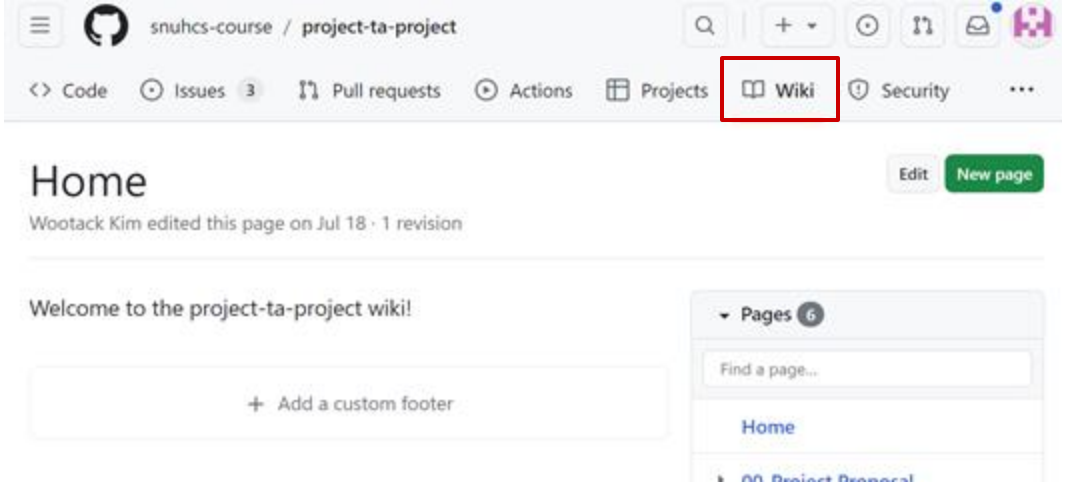
- Discuss the following with TA every Thursday
  - Working demo
  - Design documents
  - Schedule
  - Git usage
  - Metrics
  - Plus anything your team/TA wants to discuss
- Submit everything at the end of each iteration via git

Working Demo

- At the end of each iteration, submit a working demo.
- Submission Instructions:
  - Create a separate branch for your demo (iteration-#-demo)
  - Include a README.md in the root of that branch containing:
    - How to run your demo including clear setup and execution instructions
    - What your demo demonstrates
  - A short demo video showcasing the key functionalities
    - Video doesn’t have to be fancy

Project Code and Documents

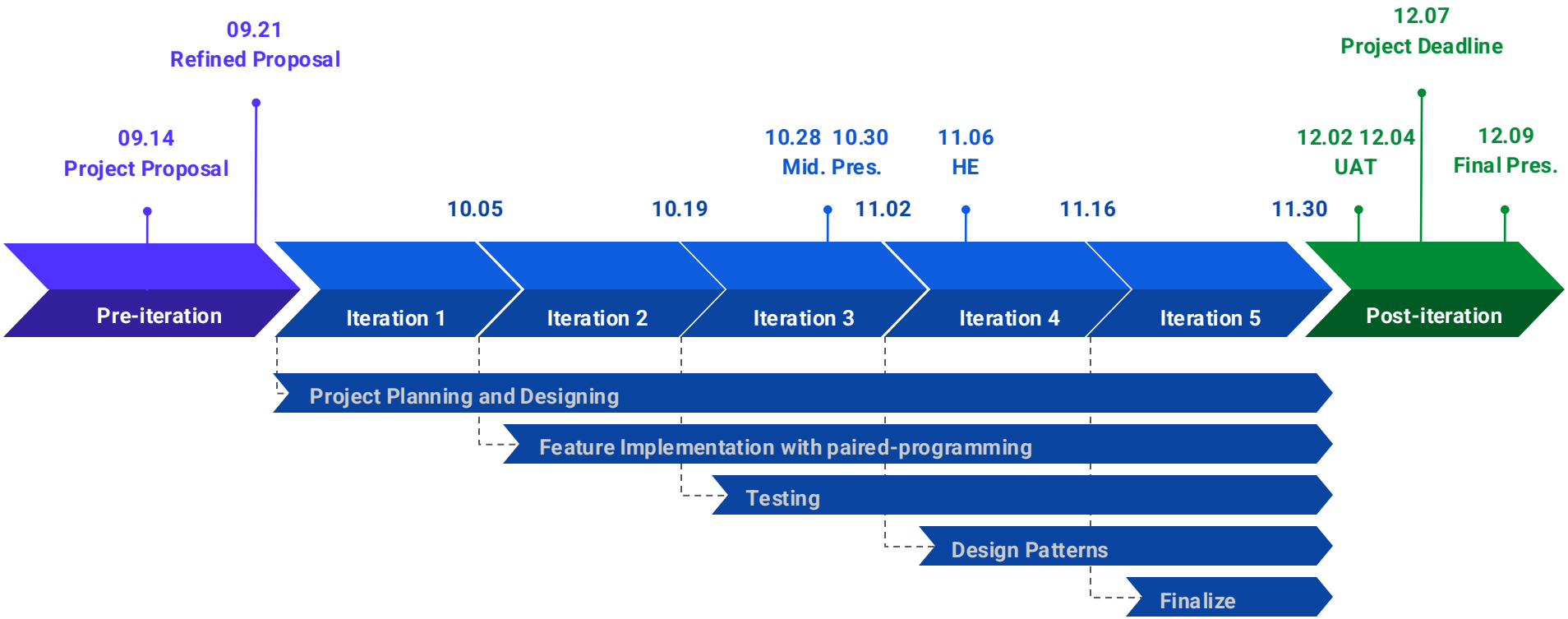
- All your code will be managed via GitHub
- All project documents will be managed in the Wiki page
- Code and documentation should be regularly updated



Project Process Overview

- We will use combination of agile processes
  - XP practices
    - Collective code ownership, coding standards
  - RUP/Scrum-like practices
    - 5 Iterations, use case-driven development
  - Common
    - Consider specified milestones
    - Track metrics
    - Use GitHub and Wiki
    - Role rotations
      - 1 project manager and 4 programmers per iteration

Project Process Overview



Pre-Iteration: Project Proposal

- 2 pages including images
- Refer to the proposal guideline & example for more details
- Timeline:
  - 09/14 23:59:59 - proposal deadline
  - 09/15 23:59:59 - TA feedback
  - 09/21 23:59:59 - refined final proposal deadline
- Upload to GitHub repository (Wiki) or submit to TAs via eTL

Iteration 1: Planning and Designing

- Start project planning and software design
- Your team needs to prepare the following documents
  - Project Schedule (Google Sheet)
  - Requirements & Specifications (GitHub Wiki)
  - Design Documentation (GitHub Wiki)
  - Meeting Logs (GitHub Wiki)
- Demo your wireframe mockup or a very simple feature (like login)

Iteration 2: Start Implementation

- Start major implementation! (with YOUR GitHub id)
- Avoid re-creating the git repository so make sure to avoid catastrophic commits/pushes!
- Log commit status so that PM can track progress
- PM updates the schedule sheet accordingly
- Update design docs

Iteration 3: Start Testing

- Continue implementing features
- Must start unit testing!
  - Log test coverage
  - Fix bugs when testing!
- PM: must start integration testing!
  - Track tested user stories
- Be prepared for the midterm presentation and demo
- Get ready for the heuristic evaluation for usability testing

Iteration 4: Apply Design Patterns

- Continue implementing and testing
- Apply design patterns to your project
- Specify in Design Documentation which design pattern you apply, why and how
- Most parts of the app should be ready and usable

Iteration 4: Heuristic Evaluation

- PM (+1 person) explains to other teams how to test
- Other team members evaluate other teams' applications
- After evaluation, organize usability issues found by other team members, and propose solutions to apply

Iteration 5: Finalize App and Project!

- Finish any implementations left
- Refactor and clean code
- Deploy your app for User Acceptance Testing (UAT)
- Make UAT test plan and discuss with TA
- Be ready for the final presentation

Post-Iteration: User Acceptance Testing (UAT)

- Test plans
  - Delimit success criteria for each feature (connected to a user story)
  - Testing should accept FREE INPUT to avoid hard-coded test paths
  - TA will discuss whether test plans are challenging enough
- Test results
  - Organize test results and analyze them in docs
- UAT is the final test by the customer in practice
  - We will give few more days to fix minor issues and wrap up
  - This is for learning experiences, different from the practice

Post-Iteration: Final Deadline

- Final deadline to submit presentation slides
- If you want (and can), you can fix some minor bugs to meet the feedbacks from UAT
- Any code pushed after deadline will NOT be considered

Project Documents Overview

Project Schedule: Why Use It? (1/2)

- Clear Vision
  - A well-planned project schedule helps team members understand the project's goal and the expected end product, aligning them to a shared vision
- Allocation and Utilization
  - By planning 'who will do what', 'when', and 'how much time it will take', teams can ensure that everyone contributes effectively and efficiently

Project Schedule: Why Use It? (2/2)

- Risk Mitigation
  - By planning in advance, teams can identify potential risks and develop backup plans
- Inspection and Feedback
  - By setting milestones and deadlines, a project schedule offers a measurement tool to track progress and identify deviations early

Project Schedule: Misc.

- Use the google spreadsheet template to track the schedule and share it with your team's TA
- Plan till the end – of course you can revise, but still plan!
- Consider the aforementioned project milestones when you come up with your project schedule
- PM updates the project schedule regularly upon task completion and iteration starts

Project Schedule: Template (1/3)

- Team & Project Overview
  - Set your project's goal, motivation, and expected end product
  - Define 'House Rules' and establish work norms
    - Collective goal of the team such as actual deployment to the Android market, just passing the course, etc.
    - No coding after midnight
    - Don't be late to the meetings

Project Schedule: Template (2/3)

- PM Rotation Schedule
  - 1 project manager and 4 programmers per iteration
    - Project manager: schedule tasks for the iteration, assign members, check their progress, test, etc.
    - Feel free to have another role. But make sure everyone contributes to implementation except for the PM.
  - Each team member will experience to be a PM
  - Keep the roles within each iteration

Project Schedule: Template (3/3)

- Project Task Breakdown
  - A detailed task breakdown helps understanding the project's scope and dependencies
- Iterations & Task Assignments
  - Treat each iteration as a self-contained project with a goal and assignments
  - Estimate how much resources each task will take and plan effectively

Requirements & Specifications (1/3)

- Overview of how the app interacts with the outside world
- Focuses on what users need and not on implementation
- Contains:
  - Project Abstract: summary of the software, < 200 words
  - Customer: brief description of the target customer
  - Competitive Landscape
    - Identify the competitors of the market
    - List the main ways in which your project is different

Requirements & Specifications (2/3)

- Contains (cont'd):
  - Functional Requirements
    - User Stories & User Acceptance Criteria
    - Short description of a user's interaction with the system
    - Example behavior of the system written from a user's POV
    - Must include Acceptance Tests
    - Minimum 5, which cannot be changed after Iteration 3
  - Non-Functional Requirements
    - Not mandatory logic but essential for high usuability
    - E.g., response latency, scalability, security

Requirements & Specifications (3/3)

- Contains (cont'd):
  - UI Requirements
    - Show how screens will look like and how they interact
    - Sketches or mockups for main parts of the user interface
    - You can use online tools: figma

Design Documents (1/2)

- All the technical documents to guide implementation
- Should serve as the blueprints for all developers
- Contains:
  - System Architecture
    - Major pieces of your system and how they interact
    - Use graphical notations such as UML
    - Include major interfaces between components
    - Use standard architectural elements (e.g., model-view-controller, client-server, …) - some more here

Design Documents (2/2)

- Contains (cont'd):
  - Design Details
    - Class diagram for your system
    - The database schema
    - App-specific details such as algorithms, business logic, DNN models
  - Testing Plan (from Iteration 3 onwards)
    - For unit testing, UI testing, and integration testing
    - Specify which tools you plan on using for testing

Meeting Logs

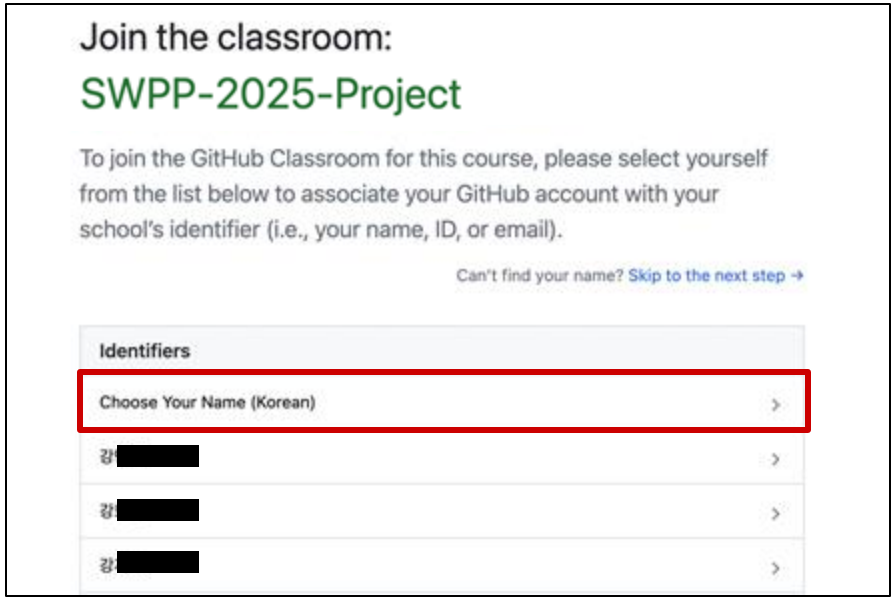
- Leave the daily standup meeting log
- Simply leave the log in the git wiki
- Communicate with your team daily
- Free format that contains time and key discussion points

Team Exercises

5 mins

Team Exercise 1: Join GitHub Classroom

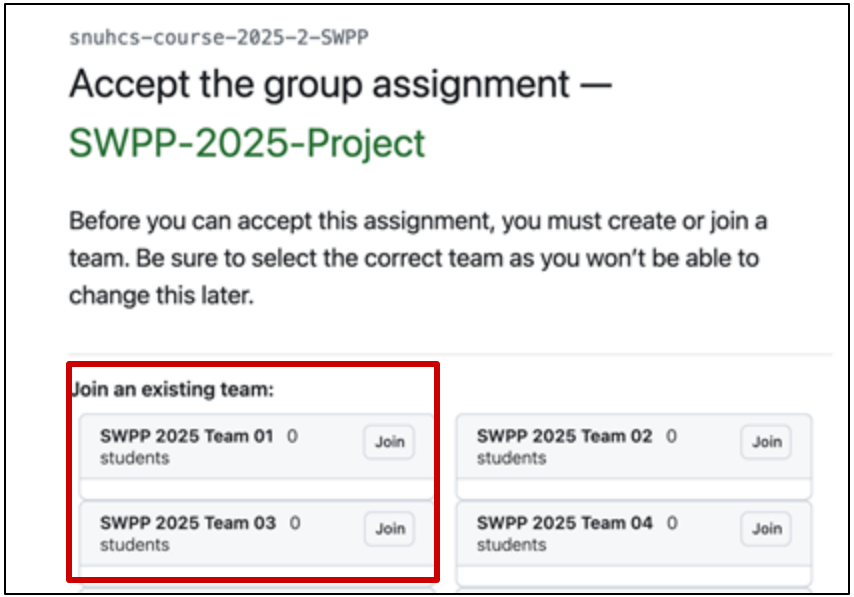
- Follow the link and select your name
  - GitHub Classroom : https://classroom.github.com/a/5AOQKrq4

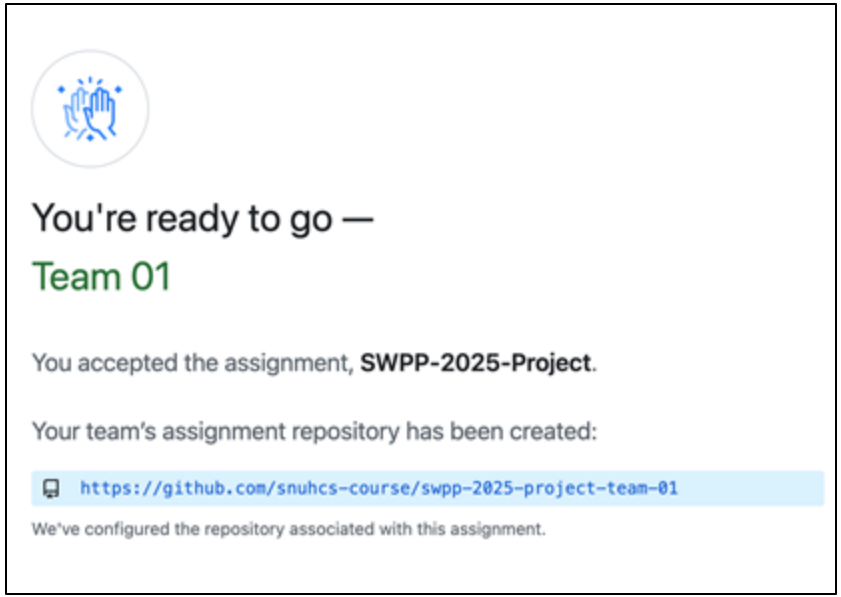


5 mins

Team Exercise 1: Join GitHub Classroom

- Join to your assigned team

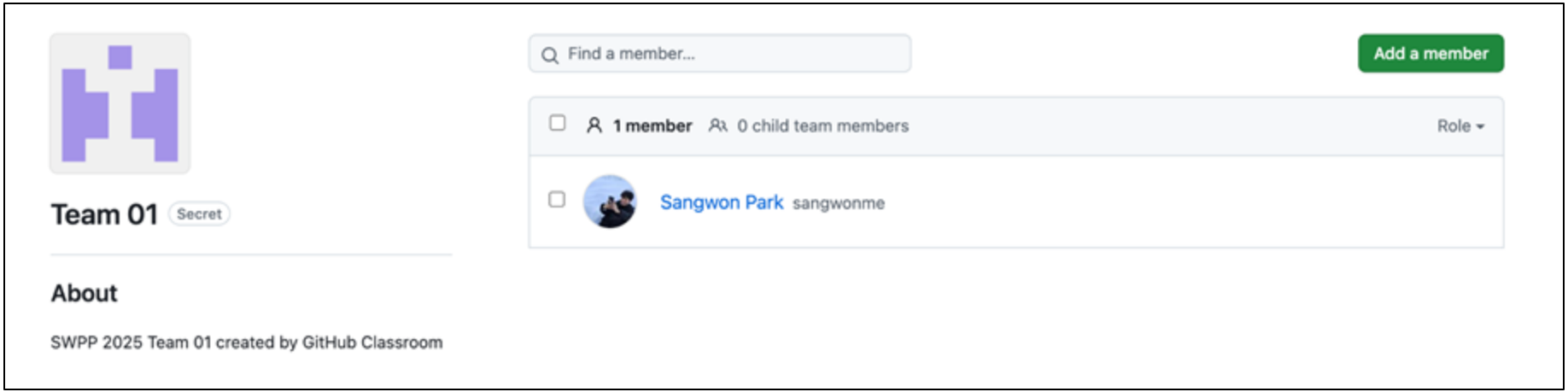




5 mins

Team Exercise 1: Join GitHub Classroom

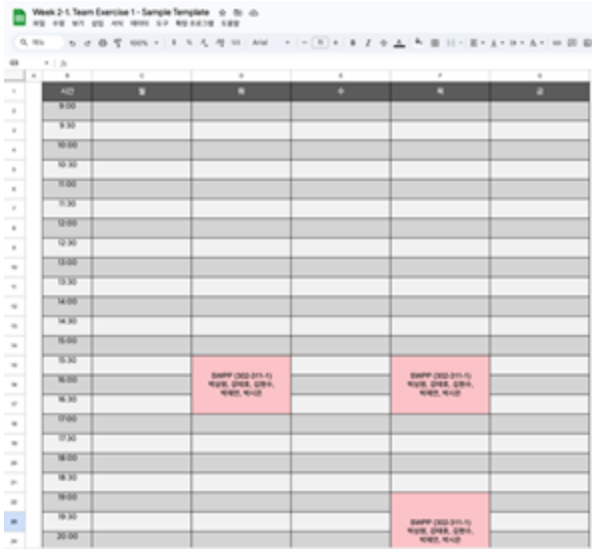
- If you joined to wrong team, let TA knows
  - Team: https://github.com/orgs/snuhcs-course/teams



10 mins

Team Exercise 2: Create Shared Timetable

- Create a combined timetable of all your members using google spreadsheet. Here is a sample template
- Share the spreadsheet with your team's TA via Slack



10 mins

Team Exercise 3: Introduce your group!

- Create a short presentation with the following information:
  - Your team's name
  - Your team's goal (very important!)
  - A screen capture of your team's common timetable
  - Brief information about each member of the team
  - Anything else you would like to share
- Submit the slides on eTL until the end of class
- Lets hear from teams on Thursday.

Thank You.
Any Questions?