CPEN 321 Software Engineering
Winter 2023

M4: MVP (Friday, Oct 27, 9pm PT)

The deliverable for this milestone is your project **MVP** (minimal viable product). In this milestone, all project use cases (functional requirements) have to be implemented, be functional, and demonstratable.

To communicate with the backend server, your app should use HTTPS rather than HTTP.

- Ensure that your server is configured to listen to HTTPS requests. To deal with SSL handshake issues, you can use self-signed certificates generated using your server's public IP or certificates from popular Certificate Authorities, such as Let's Encrypt.

You **can** use assistive AI technology (namely, ChatGPT 3.5) when working on this assignment. No points will be deducted for documented use. However, undocumented use will be considered academic misconduct and will be treated accordingly.

One of the educational objectives of this assignment is to explore the usefulness of AI tools in software engineering processes. As such, you must document and critically analyze all usages of ChatGPT during the process of working on this assignment in a systematic way described below. This analysis must be submitted as part of the assignment and will be graded. Using ChatGPT 4 or any other technology is not allowed, for fairness (so that all students will get the same level of support).

Submission: The submission will include three parts.

PART I. This part of the submission is a project report. The report must include:

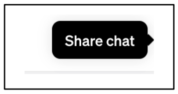
1. Changes in the project scope, requirements, and design, since the corresponding milestones **[8 points]**.
   - If no changes were made, state "None". However, note that **mismatches** between your requirements/design and your implementation will cause you to lose marks.
   - If there are changes, describe each change by including:
     - The new artifact (a use case diagram, a functional requirement specification, a non-functional requirement, a new module, interface, design diagram, etc.), as required in the previous milestones.
     - The reason behind the change. Please make sure the justification is clear and reasonable.
2. Contribution of each group member to the work done for this milestone, in 1-2 sentences per member **[2 points]**.
3. A public IP of your back-end server and a list of backend APIs that the back-end exposes to the front-end, together with the API parameters **[2 points]**.
4. Which use cases implement your "complexity" idea and how **[5 points]**.

5. Reflections on using / not using ChatGPT 3.5 when working on the assignment **[20 points]**. If you did not use ChatGPT 3.5 when working on the assignment, you must answer the following questions in 1-2 paragraphs each:
    - Why did you decide not to use ChatGPT for this assignment?
    - Show concrete examples of its inadequacy.
    - Did you use any online resources (StackOverflow or similar) instead? If so, name the resources you used, explain how the resource was used, and why this is more beneficial than using ChatGPT.
    - Is there anything else you would like to share about this process?

    If you used ChatGPT 3.5, you must answer the following questions in 1-2 paragraphs each:
    - List all tasks ChatGPT helped you to achieve and the reason you decided to use it for each of these tasks.
    - How much of ChatGPT's output did you use for your implementation? (0 – 100%). Specify the total number of methods in your front-end and back-end code and the total number of methods in your code for which you rely on ChatGPT output, possibly refined. Calculate the fraction of methods created with the assistance of ChatGPT from the total number of methods.
        - In your code, you must add a comment before the name of each method to specify whether or not ChatGPT output (possibly refined) was used for implementing the method. The comment should be formatted as follows: "ChatGPT usage: Yes/Partial/No").
    - What are the advantages of using ChatGPT for each of these tasks?
    - What are the disadvantages of using ChatGPT for each of these tasks?
    - Is there anything else you would like to share about this process?
6. Names and a short description of the purpose of all ChatGPT conversations (prompts and replies) used when working on this milestone. Each conversation should be saved and uploaded as an HTML file, as specified in Part II **[2 points]**.
7. A link to the Git repository for your project **[1 point]**.
    - Please remember to give all TAs write access to the repository, as specified in the "Project Selection and GitHub Repository Setup" assignment.
    - The repository should contain two high-level folders: frontend and backend.
    - The working version of your project (MVP) should be in the latest commit of the **main** branch of the repository.
    - You can continue developing your project after the submission, but you need to use a separate branch for that.
    - Note that we will check that the date of the latest commit on the main branch precedes the date of your submission.

PART II: Submit all ChatGPT conversations (prompts and replies) that were used or attempted when working on this milestone. Each conversation should be saved and uploaded as an HTML file, using the procedure described below:

- Click on the "Share Chat" icon in the top right corner of the ChatGPT Web window.
- Click on "Copy Link". The link to the chat will be copied to clipboard.
- Open the link in a new tab of a Web browser.
- Save the page by pressing Ctrl + S.



- Name the file using the following schema: MVP_N.html, where N is the sequential number of the chat used for this milestone. For example, “MVP _2” is the second chat used for this milestone.

Assignments that relied on ChatGPT but do not declare the use and do not attach ChatGPT conversations will lose all marks for the milestone.

PART III. This part of the submission is an **APK** file. Submit an Android release APK build of your mobile application and make sure your back-end server is up and running until the grades are released **[60 points].**

Before submission, ensure that:

- The app works on a Google Pixel 3 emulator running Android Q (API 29).
  - Build the APK and install it on the emulator by dragging and dropping the APK onto the emulator screen.
- Sign-in works properly for the release APK version that you submit.
  - E.g., if you use Google sign-in, you will need to add the SHA-1 of the public key used to sign the submitted APK (i.e., the developer’s key for the signed apk) on the Google API Console.
- Verify that all the implemented functionality is working correctly.
- The back-end server is up and running from the submission deadline until the assignment grades are posted on Canvas. If the server is unreachable when a TA grades your assignment, you will lose all marks for this part.

The MVP implementation will be graded based on its scope and completeness.

Good luck!