Office Hours Tracker: Project Report

1. **Two paragraph summary of the project *as* implemented, including the main customer need and how the application meets it, including who the stakeholders are. This will contrast to what you wrote in Iteration 0.**

   Our customer came to us with a few main requirements: Students needed to be able to log attendance at office hours for a specific class, TA's had to be able to record their attendance for billing purposes, and this data all needed to be accessible.

   To start, our team uses Google OAuth to allow users to login with an email address. This email address is then checked against a file to see if it is a known TA or Admin email (this is editable by the admin). If they are not on the lists, they will go to a student page where they can first select the class, and then record their attendance. If they are a TA, they go to the TA page where they can record that they attended their shift. If they are an admin, they go to the admin page which contains buttons to download the student or TA attendance data. Admins can also see the student attendance data visualised in a chart on their homepage. There is also a button to another page so that they can upload TA and admin email lists. Admins also have the ability to edit which courses are shown to the students for recording their office hour visits.

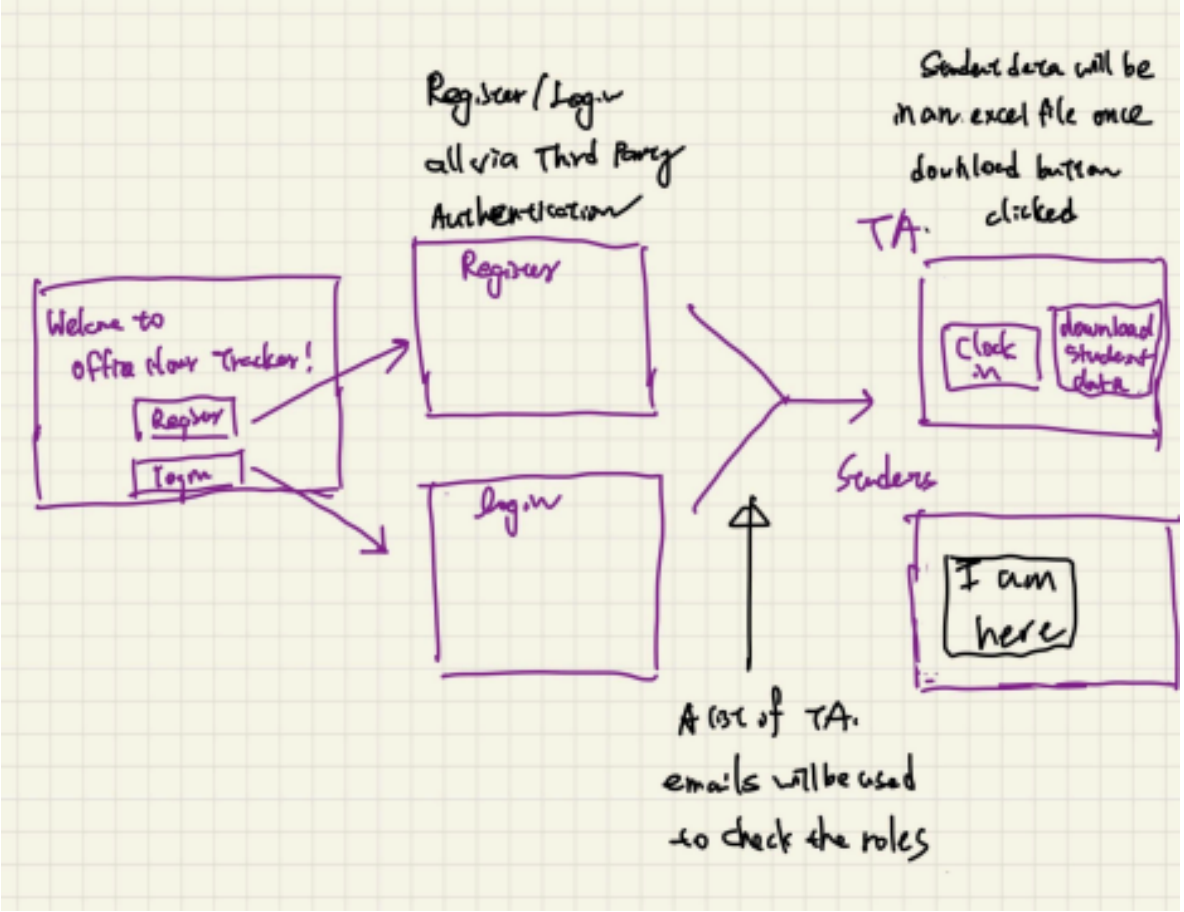
2. **Description of *all* user stories (including revised/refactored stories in the case of legacy projects). For each story, explain how many points you gave it, explain the implementation status, including those that did not get implemented. Discuss changes to each story as they went. Show lo-fi UI mockups/storyboards you created and then the corresponding screen shots, as needed to explain to stories.**

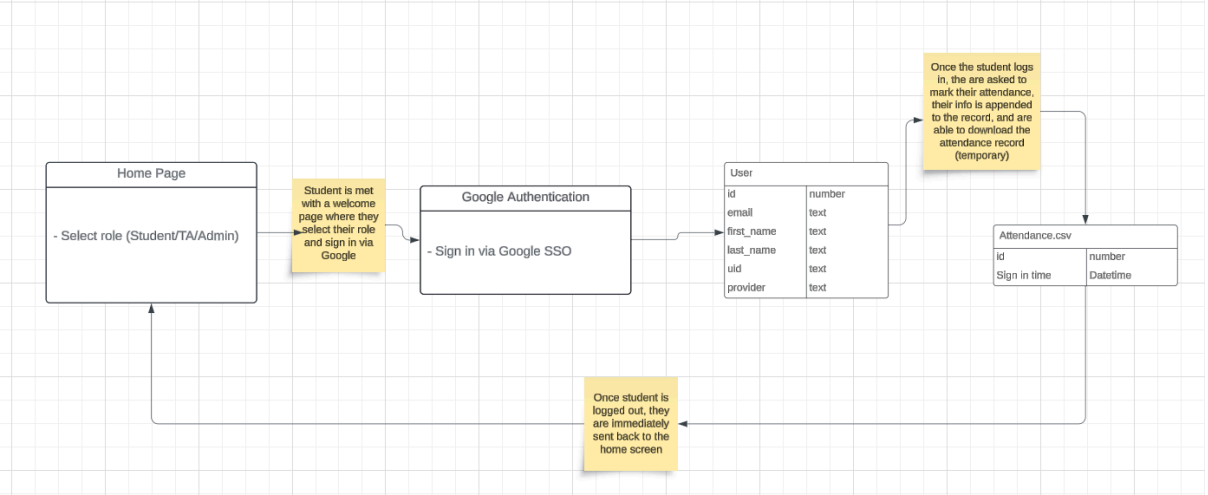
   Sprint 1

   —----------

Story	Status	Story Points
As a <student/TA/Admin> I want to <register>, so that <system knows who I am>. Changes/Reasons: We merge this task with the login, for register and login means the same in OAuth	Completed	2
As a <student/TA> I want to <show I am here>, so that <system can record my attendance>.	Completed	2

As a <TA/Admin> I want to <download> the number of students per hour so that <we can use the records for scheduling purposes>.	Completed	2
As a <student>, I want <each page to have a well-designed and intuitive UI>, so that <I can easily understand the purpose and content of each page>.	Completed	2
As a <student>, I want the < the navigation between pages to be seamless>, so that <I have a smooth and enjoyable user experience> Changes/Reason: Our team has designated the scrum master and product owner to primarily handle documentation and testing. To support their efforts, we have assigned a dedicated person to assist with testing to ensure quality.	Completed	2



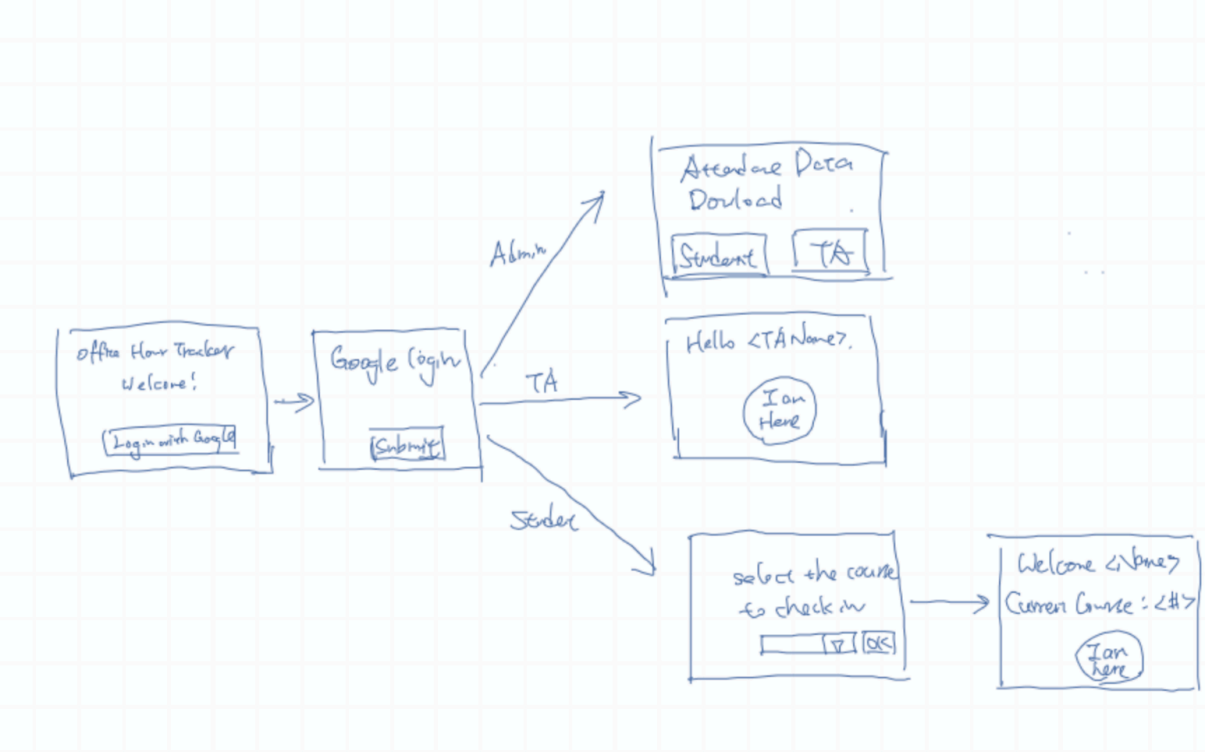


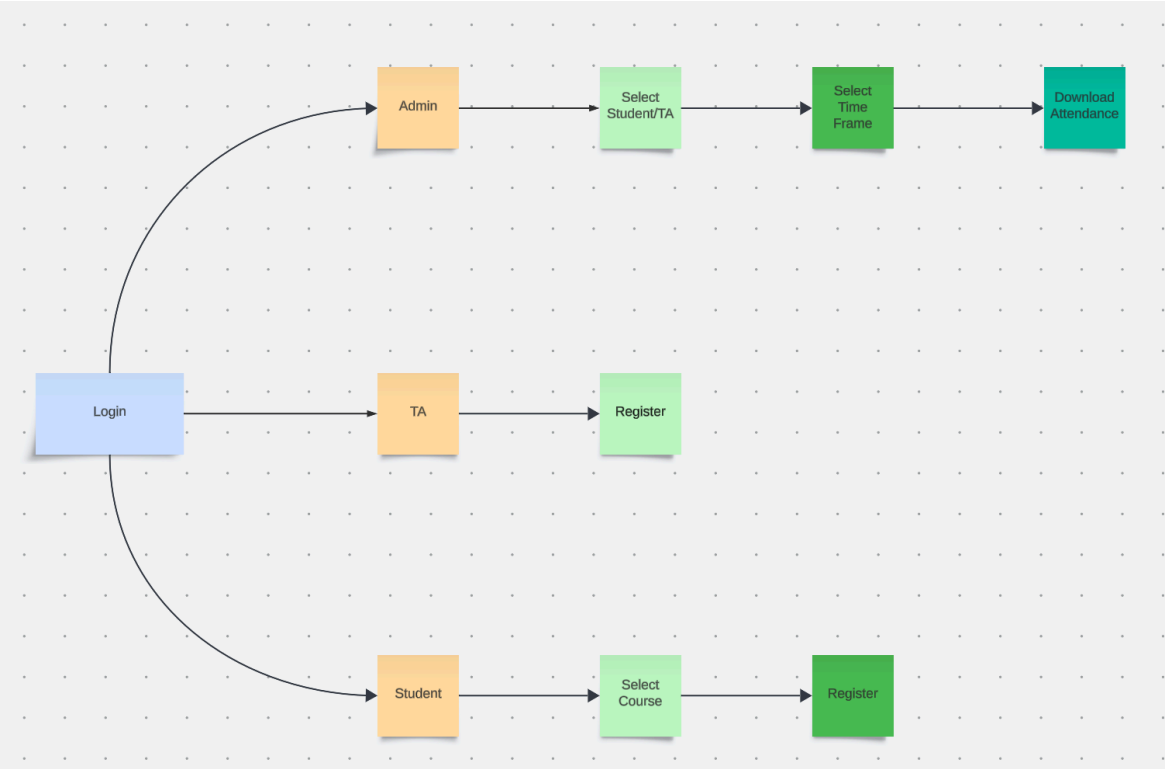
Sprint 2

—----------

As a <scrum master>, I want to <arrange daily meetings and remove difficulties encountered by teammates>, so that <the developers will have reasonable progress during the sprint>	Completed	2
As a <Student/TA>, I want to see <already registered when marking attendance repeatedly>, so that <a new entry is not added to same time slot.>Change/Reason: To ensure consistency, we move the database creation (TAattendance) to Gorangi.	Completed	2
As a <student>, I want to <select from a dropdown of course>, so that <I know which course I am marking attendance for>	Completed	2
As a <Admin/TA/Student>, I want to <know my role after login>, so that <I can check if I am registered correctly> Change/Reason: The current implementation doesn't store the TA/admin information into file, instead, certain files are queried to determine roles.	Completed	2

As a <product owner>, I want to <prioritize the story that should be addressed in this sprint>, so that <the client is able to use the product after this sprint.>	Completed	2
As a <TA> I want to <show I am here>, so that <system can record the attendance of me>	Completed	2
As an <Admin>, I want to <download the attendance data from TA and student>, so that <I know the number of person joined the slot> Change/Reason: The actual implementation is changed during the sprint because we find creating two buttons for TA and student download will conflict with the option of choosing data range. The current UI is shown below. Information in three boxes have to be specified before hitting the download button.	Completed	2



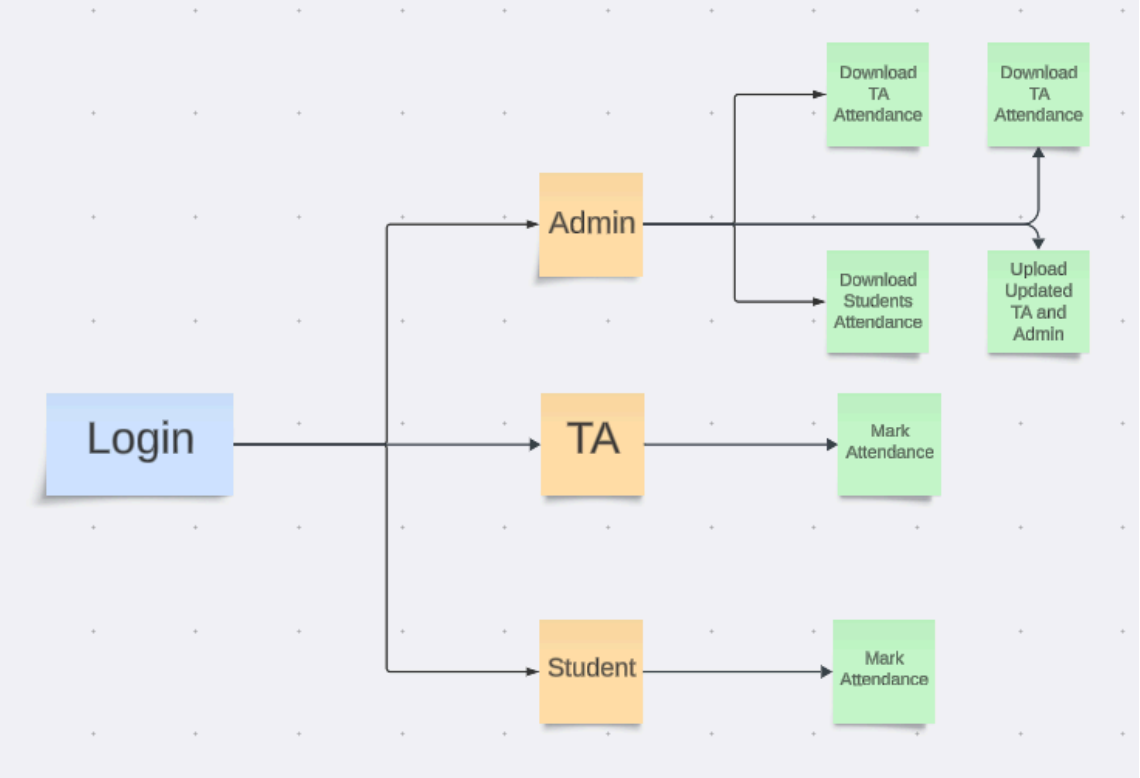


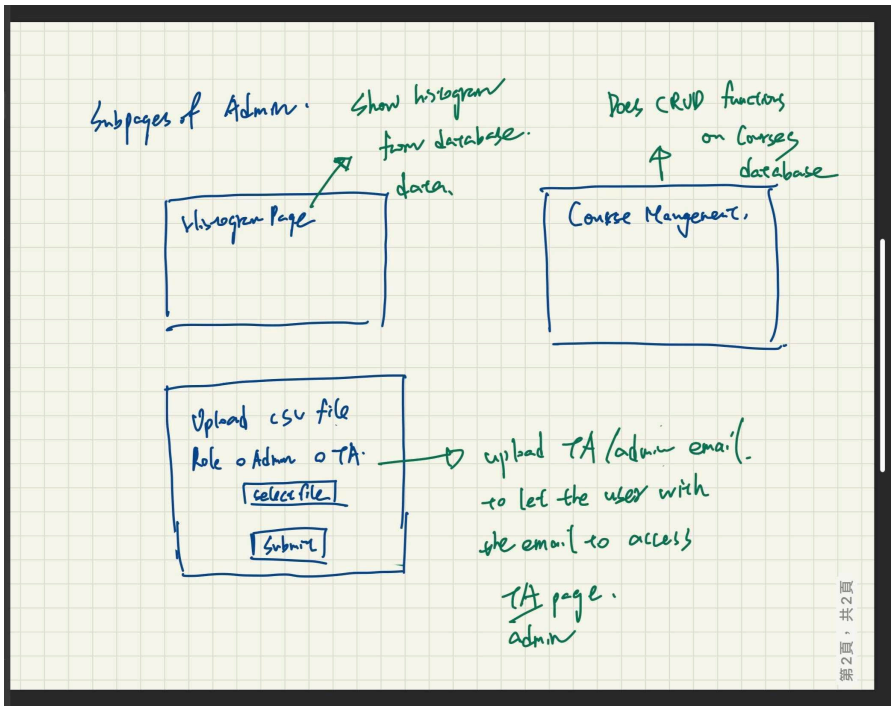
Sprint 3

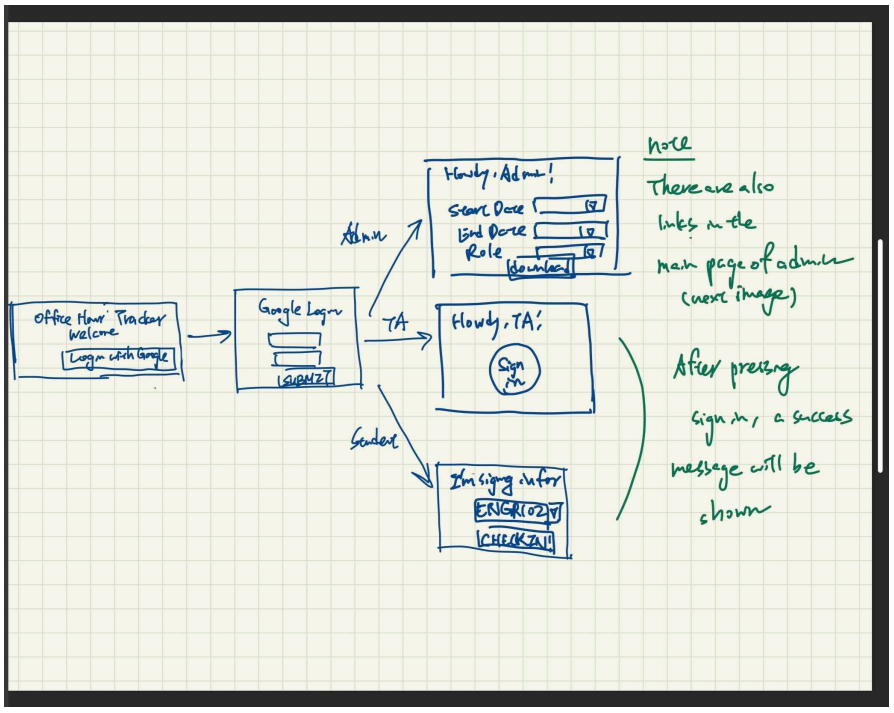
———————————

Story	Status	Story Points
As an <Admin>, I want to <see attendance graphs>, so that I can understand attendance trends.	Completed	2
As a <product owner>, I want to <prioritize the story that should be addressed in this sprint>, so that <the client can use the product after this sprint.>	Completed	2
As an <Admin>, I want to <see student attendance in UI>, so that I can have a record of attendance for analysis in UI.	Completed	2
As an <admin> I want to <upload a Admin/TA email file>, so that <I can change the Admins/TA if needed.>	Completed	2
As an <admin> I want to <change the available courses>, so that <courses information can be updated each semester>	Completed	2

As a <User/TA>, I want to <make the GUI more intuitive>, so that <users can navigate the application easily and have a positive user experience>	Completed	1
As an <Admin>, I want to <see the list of TA checked in>, so that <the admin can use it to process payroll> Change/Reason: The person doing this task accidentally makes a list of people appending to specific time slots instead of showing people's check-in time. Should be rectified in the next sprint.	Completed	1
As a <scrum master>, I want to <arrange daily meetings and remove difficulties encountered by teammates>, so that <the developers will have reasonable progress during the sprint>	Completed	2



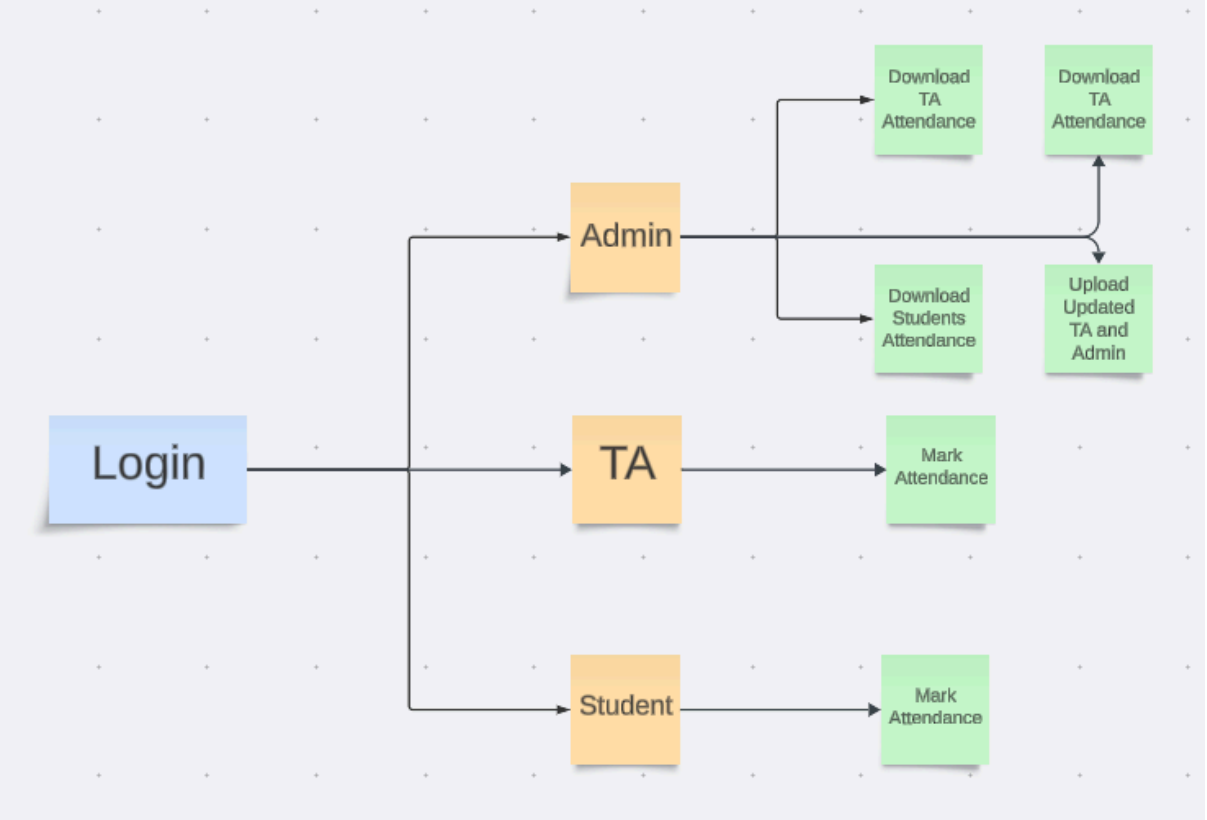

Subpages of Admin.
Show histogram from database.
data
Histogram Page
Does CRUD functions on Courses database
Course Mangement.
Upload csv file
Role o Admin o TA.
select file
Submit
upload TA/admin email to let the user with the email to access TA page. admin
第2頁，共2頁


Office Hour Tracker
welcome
Login with Google
Google Login
SIGN IN
Admin
TA
Student
Howdy, Admin!
Start Date
End Date
Role
Howdy, TA!
Sign in
Im signing infor
ENGR102
CHECK IN!
Note
There are also links in the main page of admin (next image)
After pressing sign in, a success message will be shown

Sprint 4
—----------

Story	Status	Story Points
As an <Admin>, I want to <view the data in both histogram and table form>, so that <I can easily interpret the data> (*) ii) Make the histogram show data in CST timezone. Change/Reason: Fix UTC and CST time zone differences	Completed	2
As an <Admin>, I want to <be able to navigate the portal effectively>, so that <better view relevant information>	Completed	2
As an <Admin/TA/Student>, I want to <sign in>, so that <no bugs arise in the process> Change/Reason: Merge two tasks into one because they convey the same meaning	Completed	2
As an <Admin>, I want to <ensure the student csv functions work accordingly>, so that <I can gauge the attendance for each day> (*) ii) Make the csv file show data in CST timezone. Change/Reason: We observe that the database writes time in UTC timezone, so one has to do some post-processing to get data in CST time.	Completed	2
As an <TA>, I want to <use the TA portal>, so that <I can ensure that I am signed in to my shifts> (*) ii) Make the TA csv file show data in CST timezone. Change/Reason: We observe that the database writes time in UTC timezone, so one has to do some post-processing to get data in CST time.	Completed	2
As a <product owner>, I want to <prioritize the story that should be addressed in this sprint>, so that <the client can use the product after this sprint.>	Completed	2

As a <scrum master>, I want to <arrange daily meetings and remove difficulties encountered by teammates>, so that <the developers will have reasonable progress during the sprint>	Completed	2



3. **List who held each team role, e.g. Scrum Master, Product Owner. Describe any changes in roles during the project**.
    - Sprint 1:
        - **Project Owner**: Bryson Brown
        - **Scrum Master**: Ethan Tran
        - **Developers**: All the other people
    - Sprint 2:
        - **Project Owner**: Wei Chein Kao
        - **Scrum Master**: Chen Hung
        - **Developers**: All the other people
    - Sprint 3:
        - **Project Owner**: Gourangi Sanjay Taware
        - **Scrum Master**: Nhat Nguyen
        - **Developers**: All the other people
    - Sprint 4:
        - **Project Owner**: Ethan
        - **Scrum Master**: Nazar Oladepo
        - **Developers**: All the other people

4. For each scrum iteration, summarize what was accomplished and points completed.

Sprint 1:

- Added Google OAuth to handle accounts
- Added main page and CSS
- Added login feature
- Added database to store marked attendance
- Added attendance marking
- Added download database function
- Added rspec and cucumber tests
- Overall, completed 14 story points

Sprint 2:

- Attendance marking cannot be done twice in the same hour
- Added a dropdown for selecting which course to check in for
- Improved UI
- See list of TAs checked in and download it
- Added TA homepage and TA check in button
- Added respective rspec & cucumber tests
- Overall, completed 14 story points

Sprint 3:

- Added graphs of student data
- Added page for the Admin to edit list of people who can access the admin and TA page
- Added basic toggle functionality to let the admin download student attendance data based on time
- Added page for the Admin to edit Courses
- Added respective rspec & cucumber tests
- Overall, completed 14 story points

Sprint 4:

- Bug fixes with Google OAuth, Histogram rendering
- Make the histogram, the list under the histogram, and download toggle button work more seamlessly
- Responsiveness fixes with student page, admin TA/Admin upload features.
- Added respective rspec & cucumber tests
- Overall, completed 14 story points

5. **Include a table that tells how many stories and points each member of the team completed, in total.**

Name	Sprint number: Tasks (Points)
Chen-Hung	- Sprint 1: As a <student/TA/Admin>, I want to <login>, so that <I can enter the system>. (2) - Sprint 2: As a <scrum master>, I want to <arrange daily meetings and remove difficulties encountered by teammates>, so that <the developers will have reasonable progress during the sprint> (2) - Sprint 3: As an <Admin>, I want to <see attendance graphs>, so that I can understand attendance trends. (2) - Sprint 4: As an <Admin>, I want to <view the data in both histogram and table form>, so that <I can easily interpret the data>
Gourangi	- Sprint 1: As a <student/TA> I want to <show I am here>, so that <system can record the attendance of me>. (2) - Sprint 2: As a <Student/TA>, I want to see <already registered when marking attendance repeatedly>, so that <a new entry is not added to same time slot.> (2) - Sprint 3: As a <product owner>, I want to <prioritize the story that should be addressed in this sprint>, so that <the client can use the product after this sprint.> (2) - Sprint 4: As an <Admin>, I want to <be able to navigate the portal effectively>, so that <better view relevant information> (2)
Bryson	- Sprint 1: As a <product owner>, I want to <prioritize the story that should be addressed in this sprint>, so that <the client is able to use the product after this sprint.> (2) Note: The original table in sprint 1 retrospective mentions that the person is in charge of cucumber test. Here we try to make the task assignment more clear by including role as product owner. - Sprint 2: As a <Admin/TA/Student>, I want to <know my role after login>, so that <I can check if I am registered correctly> (2) - Sprint 3: As an <admin> I want to <upload a Admin/TA email file>, so that <I can change the Admins/TA if needed.> (2) - Sprint 4: As an <Admin/TA/Student>, I want to <sign in>, so that <no bugs arise in the process> (2)
Nhat	- Sprint 1: As a <student>, I want the < the navigation between pages to be seamless>, so that <I have a smooth and enjoyable user experience>: subtask-Create database for User (2) Note: The work mentioned at the sprint 1 retrospective is actually a division of a story, so we classify it under a subtask.

	- Sprint 2: As an <Admin>, I want to <download the attendance data from TA and student>, so that <I know the number of person joined the slot> (2) - Sprint 3: As a <scrum master>, I want to <arrange daily meetings and remove difficulties encountered by teammates>, so that <the developers will have reasonable progress during the sprint> (2) - Sprint 4: As an <Admin>, I want to <ensure the student csv functions work accordingly>, so that <I can gauge the attendance for each day> (2)
Franklin	- Sprint 1: As a <TA/Admin> I want to <download> the number of students per hour so that <we can use the records for scheduling purposes> (2) - Sprint 2: As a <product owner>, I want to <prioritize the story that should be addressed in this sprint>, so that <the client is able to use the product after this sprint.> (2) - Sprint 3: As an <admin> I want to <change the available courses>, so that <courses information can be updated each semester> (2) - Sprint 4: As an <TA>, I want to <use the TA portal>, so that <I can ensure that I am signed in to my shifts> (2)
Ethan	- Sprint 1: As a <scrum master>, I want to <arrange daily meetings and remove difficulties encountered by teammates>, so that <the developers will have reasonable progress during the sprint> (2) Note: The original table in sprint 1 retrospective mentions that the person is in charge of rspec test. Here we try to make the task assignment more clear by including role as scrum master. - Sprint 2: As a <student>, I want to <select from a dropdown of course>, so that <I know which course I am marking attendance for> (2) - Sprint 3: As an <Admin>, I want to <see student attendance in UI>, so that I can have a record of attendance for analysis in UI (2) - Sprint 4: As a <product owner>, I want to <prioritize the story that should be addressed in this sprint>, so that <the client can use the product after this sprint.>. (2)
Nazaro	- Sprint 1: As a <student>, I want the < the navigation between pages to be seamless>, so that <I have a smooth and enjoyable user experience>: subtask-Create database for User: subtask - GUI styling (2) Note: The work mentioned at the sprint 1 retrospective is actually a division of a story, so we classify it under a subtask.

	- Sprint 2: As a <TA> I want to <show I am here>, so that <system can record the attendance of me> (2) - Sprint 3: As a <User/TA>, I want to <make the GUI more intuitive>, so that <users can navigate the application easily and have a positive user experience> (1) - Sprint 3: As an <Admin>, I want to <see the list of TA checked in>, so that <the admin can use it to process payroll> (1) - Sprint 4: As a <scrum master>, I want to <arrange daily meetings and remove difficulties encountered by teammates>, so that <the developers will have reasonable progress during the sprint> (2)

6. **List of customer meeting dates, and description of what happened at the meetings, e.g. what software/stories did you demo.**

Sprint 1: Wed Sep 25

Our group met with the client to discuss the app requirements for tracking office hours for students and admins (TAs/instructors). The app will allow students to mark their attendance. TAs can clock their shifts in the web application and mark their attendance. Admin can view Excel reports for payroll and analytics. Users can switch between student and admin roles depending on their class.

Demos: None

Sprint 2 Wed Oct 9

During the meeting, we started by giving a brief overview of the progress that we made as well as a demo of the current features of the project. We then asked our client for feedback on the product, and she liked the format of the page currently. We then clarified the full scope of the project regarding the classes that will be using it. Based on this, we will be continuing on with our admin pages and TA pages as we planned, but we will modify the student page to select the specific class the student is there to get help for. We also planned on giving the client a version of the application to be able to test at the end of sprint 2, with the improvements to the functionality being added modularly in sprints 3 and 4.

Demos: We demoed the login page and the student page, as well as the student attendance feature

Sprint 3 Wed Oct 23

During the meeting, we started by giving a brief overview of our progress and a demo of the current features of the project. We then asked our client for feedback on the product, and she liked the format of the page currently. We then clarified the full scope of the project regarding the classes that will be using it. Based on this, we will continue with our admin pages and data visualization as planned, but we will modify the TA CSV page to add TA names. We also planned on giving the client a version of the application to test at the end of the sprint, with the improvements to the functionality being added modularly in sprints 3 and

Demos: We demoed the student, ta, and admin pages as well as the TA ability to record attendance.

Sprint 4 Wed Nov 6

During this meeting of our final sprint, we review our progress thus far and address any bugs our application may have. Based on these last few issues, we shall assign bug fixes to the elements that a member has worked on the most and modify our application to be more accessible to the users. We have not met with the client due to timing issues, but we plan on having a complete application by the end of this sprint to present to the client in the next meeting.

Demos: We showed the near-complete product, showing the new graphs on the Admin page and the new download-window feature.

7. **Explain your BDD/TDD process, and any benefits/problems from it.**

   The BTD/TDD process was to have 5 people work on implementing the main features of the sprint, then have the remaining 2 people build tests. The features of the sprint would ideally be finished 2-3 days before the MVP is due which gives time for the 2 people to write tests.

   The benefit was that everyone had roughly an equal amount of work, and that by adding test cases after the feature we didn't have to constantly update the test cases during the sprint due to small adjustments or changes. Another benefit is that every feature is validated by at least 2 people, one for creating the feature and one for testing.

   The issue with this system was that it creates a bottleneck point where everyone must submit their sprint before tests could be created. This was mitigated slightly by having test writing be to cover the current unrefined code and updated later once all code is in. Another issue was that because the scrum master and product owner wrote the test, test writing is passed around each sprint which makes it hard to build momentum to work in a specific area of the codebase.

8. **Discuss your configuration management approach. Did you need to do any spikes? How many branches and releases did you have?**

   In sprint 2, the team faced a disagreement on how to scale the database. Some team members proposed using multiple associations to connect different databases, while others raised concerns about the potential complexity it could introduce for future development. To resolve this, we conducted a spike by asking a team member to assess the difficulty and implications of adding multiple associations. Based on the insights gained, we were able to reach a consensus on the best approach to proceed

   In our project, we completed a total of four releases, each aligned with the sprint MVP deadlines. To ensure clear versioning and traceability, each release was tagged in the repository

   For branches in the repository, we actively maintained only the production branch to streamline our workflow. Major changes were developed on separate branches, and developers submitted pull requests for code review before merging changes into the production branch. After the merge, developers could delete their branches, ensuring the repository remained clean and focused.

   This approach allowed us to ensure that unstable code was never pushed to production, preserving the integrity of the codebase. Additionally, by minimizing the number of active branches, we reduced the overhead of maintaining multiple branches, allowing the team to focus on delivering features and fixes efficiently.

9. **Discuss any issues you had in the production release process to Heroku.**

Previously, there was an incident where attempting to access the deployed website resulted in a dyno= connect= service= status=503 error message, making the site inaccessible. At the time, the cause was unclear, and upon reviewing the logs, no one appeared to have deployed an incorrect version. This left us without a clear course of action. After researching the error, we discovered that the issue was due to exceeding the limits of the original plan. We resolved it by changing the plan from “Essential0” to “Eco Subscription,” and the problem hasn’t recurred since. However, what remains puzzling is that the “Essential0” plan’s quota should not have been fully used up under normal circumstances.

10. **Describe the tools/Gems you used, such as GitHub, CodeClimate, SimpleCov, and their benefits and problems.**

    GitHub - We used GitHub to manage code pushes and it worked really well. We tried using it in the first sprint to manage user stories, and that did not work well because there is not way to assign and keep track of points for the stories.

    Jira - After GitHub Failed, we moved over to Jira in order to handle user stories. This worked very well and allowed us to have a comprehensive list of user stories and their assigned points.

    CodeClimate - Code climate requires us to upload the entire repo to it to get a score. Overall the score is good. The issue was that it requires us to be the owner of the repository which made it inconvenient to use (only the github owner can use it or we have to clone the repo which is time consuming).

    Lucid chart - When the database schema is rapidly evolving in sprint 2 and 3, we use this application to reach consensus on the correlation between different databases and the column in each database. The benefit is that it offers an intuitive GUI to let people draw boxes and lines and type text easily. We don’t experience any difficulties using this app.

11. **Make sure all code (including Cucumber and RSpec!) is pushed to your public GitHub repo.**
    All changes including Cucumber and RSpec are pushed to the public Github Repo: https://github.com/et-tran50/CSCE_606_Office_Hours_Tracker

12. **Make a separate section discussing your repo contents and the process, scripts, etc., you use to deploy your code. Make very sure that everything you need to deploy your code is in the repo. We have had problems with legacy projects missing libraries. We will verify that everything is in the repo.**

The information needed to deploy the app to heroku is on the README section of the repository.

13. **Links to your Project Management tool page, public GitHub repo, and Heroku deployment, as appropriate. Make sure these are up-to-date.**

a. **Project management tool Page:**
https://tamu-team-office-tracker.atlassian.net/jira/software/projects/SCRUM/boards/1

b. **Github Repo:**
et-tran50/CSCE_606_Office_Hours_Tracker

c. **Heroku deployment:** https://office-hours-tracker-a63f1f6d64ad.herokuapp.com/

14. **Links to your presentation video and demo video.**
https://youtu.be/lExuxv-zHY4