**Sprint 1- Documentation**

***Ceremonies (Events)-Sprint Planning, Review & Retrospective***

Sprint 1 planning- from Alessio, lecturer:

1. Installation of software necessities.
2. Definition of user stories/acceptance criteria and mock-ups.
3. Preliminary implementation of data scraping

Group Task for Sprint 1:

1. Understanding the type of data and designing database schema
2. Build the scrapers and start downloading data
3. Planning frontend designs and features (getting a mock-up/prototype)
4. Setup project tracking tools for sprint planning

Sprint Retrospective: (Start, Stop, Continue doing!)

What went well:

* From the sprint planning, team members chose tasks that aligned with their strengths as a starting point for the project.
* Successful set up of GitHub, EC2, virtual environment, Figma and Trello.
* The data scrapers were developed, and the data collection process was started.
* The types of data were analysed and understood.
* A database schema was designed that aligns with our project needs/goals.
* An initial prototype for the Web App was created.

What did not go well:

* Some of the team members were having issues with creating an instance on EC2 and could not scrape the data. Therefore, a group decision was made to leave this task to the member that already managed with this, while the other members focused on their tasks for the sprint.
* For the team member whose task was the implementation of data scraping, they encountered issues with setting up AWS RDS. There was an update from the lecturer to stop using EC2 and RDS until further notice, and to continue locally for the time being.

What can be improved:

* Effective communication and interpersonal skills as a group.
* Implement daily Scrum (or every 2 days) to quickly update each other.

Team dynamics and wellbeing:

* Workload for this sprint was uneven as one team member undertook the technical/coding aspect.
* There was a team dynamic which was directed to the appropriate parties.

***Artefacts-Sprint Backlog & Product Backlog***

User stories (12-20):

1. As a user, I want to sign up and create an account.
2. As a user, I want to log in so I can access my account.
3. As a user, I want to see a map of bike stations so that I know where bikes are available.
4. As a user, I want to check bike availability at each station so that I can plan my journey.
5. As a user, I want to see real-time weather updates so that I can decide whether to cycle.
6. As a user, I want to search for a specific bike station so that I can find one near me.
7. As a user, I want to see station details (name, available bikes, empty slots) so that I can choose a station.
8. As a user, I want to view my recent rides so that I can track my trips.
9. As a user, I want to filter stations by bike availability so that I can quickly find a station with bikes.
10. As a user, I want to favourite bike stations so that I can access them quickly.
11. As a user, I want to see historical weather data for past trips.
12. As a user, I want to receive notifications when my reserved bike is about to expire.
13. As a user, I want to see the nearest bike station based on my location so that I can find bikes easily.
14. As a user, I want to see the estimated travel time to a selected station so that I can plan my journey efficiently.
15. As a user, I want to report an issue with a bike or station so that problems can be fixed quickly.
16. As a user, I want to receive an alert when a station near me has low bike availability so that I can find an alternative station in advance.

Acceptance Criteria (clear, testable and specific for each user story):

1. **User story:** As a user, I want to sign up and create an account. -The system must provide a sign-up form with fields for name, email and password. -The email must be validated to ensure it is in the correct format. -The password must meet security requirements. -After successful sign-up, the user must receive a confirmation email with a link to verify their account.
2. **User story:** As a user, I want log in so I can access my account. -The web app should provide a registration form with fields for name, email and password. -If the user entered a wrong name, email or password (incorrect credentials) and left any required field blank, an error message should be displayed. -After successful login, the user should be directed to the homepage of the system. -The user must be able to log out from their account at any time.
3. **User story:** As a user, I want to see a map of bike stations so that I know where bikes are available. -The web app should display an interactive map showing all bike stations. -The system must retrieve and display real-time station data from the JCDecaux API. -Each bike station must be represented by a marker on the map. -The user must be able to zoom in and out on the map.
4. **User story:** As a user, I want to check bike availability at each station so that I can plan my journey. -The system must display the number of available bikes and empty slots for each station in real-time. -Clicking on a bike station marker must show a popup or sidebar with bike availability details. -The data must refresh automatically every 30 seconds to ensure real-time accuracy. -If bike availability data fails to load, the web app must display an error message and prompt the user to retry.
5. **User story:** As a user, I want to see real-time weather updates so that I can decide whether to cycle. -The system must fetch and display real-time weather data using the OpenWeather API. -The weather information must include temperature, wind speed, humidity and weather conditions (eg sunny, rainy, cloudy). -The weather data must refresh at least every 10 mins to ensure accuracy. -If the user’s location is available, the system must display weather for their current location, otherwise allow manual location entry. -The web app must display an error message if the weather data fails to load.
6. **User story:** As a user, I want to search for a specific bike station so that I can find one near me. -The system must provide a search bar where users can enter a station name or location. -The search results must display matching stations dynamically as the user types. -Clicking on a station in the search results must highlight it on the map and show its details. -If no matching station is found, the system must display a “No results found” message.
7. **User story:** As a user, I want to see station details (name, available bikes, empty slots) so that I can choose a station. -The system must display the station name, number of available bikes, and number of empty slots when a user selects a station. -Clicking on a station marker on the map must open a popup or sidebar with station details. -The station details must be updated in real-time to reflect current availability. -If the station details fail to load, the system must display an error message and prompt the user to retry.
8. **User story:** As a user, I want to view my recent rides so that I can track my trips. -The web app must display a list of recent rides, including start station, end station, date and time. -The recent rides must be stored and retrieved from the user’s account history. -Users must be able to sort or filter rides by date or station. -If no recent rides are available, the system must display a “No trip history available” message.
9. **User story-** As a user, I want to filter stations by bike availability so that I can quickly find a station with bikes. -The system must provide a filter option to display only stations with available bikes. -Users must be able to set a minimum bike availability threshold (e.g. show stations with at least 3 bikes). -The filtered stations must be visually highlighted on the map while hiding stations that do not match the criteria. -If no stations meet the filter criteria, the system must display a “No stations available” message.
10. **User story-** As a user, I want to favourite bike stations so that I can access them quickly. -The web app must provide a “Favourite” button for users to save a bike station. -Users must be able to view a list of their favourited stations for quick access. -Favourite stations must be visually highlighted on the map for easy identification. -Users must be able to remove a station from their favourites at any time.
11. **User story:** As a user, I want to see historical weather data for past trips. -The system must display weather conditions (temperature, wind speed, and status like sunny rainy) for each past trip. -The historical weather data must be retrieved based on the trip’s date and time. -Users must be able to view weather details in their trip history section. -If historical weather data is unavailable, the system must display a “Weather data not available for this trip” message.
12. **User story:** As a user, I want to receive notifications when my reserved bike is about to expire. -The system must send a notification (push, email or in-app) when the bike reservation is about to expire (e.g. 5 mins before). -Users must be able to enable or disable notifications in their settings. -The notification must include the station name, reservation expiry time, and a reminder to unlock the bike. -If the notification fails to send, the system must retry or display an alert message in the app.
13. **User story:** As a user, I want to see the nearest bike station based on my location so that I can find bikes easily. -The system must request location access from the user and display the nearest bike station on the map. -If location access is denied, the system must allow users to manually enter their location to find the nearest station. -The nearest station must be highlighted on the map and display the details like available bikes and empty slots. -The system must dynamically update the nearest station if the user moves to a different location.
14. **User story:** As a user, I want to see the estimated travel time to a selected station so that I can plan my journey efficiently. -The system must allow users to select a station from the map or search results. -The system must calculate the estimated travel time based on the user’s current location and display it clearly. -The estimated travel time must be updated dynamically if the user moves or changes their selection. -If the system is unable to calculate the travel time (due to lack of location access or data), an error message must be shown.
15. **User story:** As a user, I want to report an issue with a bike or station so that problems can be fixed quickly. - The system must provide an option for users to report an issue with a bike or station (e.g., via a "Report Issue" button on the station details page). - The user must be able to select the type of issue (e.g., broken bike, unavailable bike, damaged station) and provide additional details (e.g., description, photo). - After submitting, the system must display a confirmation message saying the issue has been reported successfully. - The reported issue must be logged in the system and accessible by administrators for review and action.
16. **User story:** As a user, I want to receive an alert when a station near me has low bike availability so that I can find an alternative station in advance. - The system must send an alert notification when a nearby station's bike availability falls below a user-defined threshold (e.g., fewer than 3 bikes available). - The alert must include station name, current bike availability, and nearby alternative stations with available bikes. - Users must be able to enable or disable this alert in their account settings or preferences. - The system must ensure the alert is sent **in real-time**, so the user can make decisions promptly.

*Product Backlog: (Paused for now)-Ask Yuhan about sign up & log in!!*

A prioritised list of work items. Type: user story, technical task (that do not deliver a direct user feature), bug fix, spike (investigation)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Priority** | **Item type** | **Item** | **Description** | **Estimation** |
| 1 | Technical Task | Software necessities & tools | Set up EC2, virtual environment, GitHub, Trello |  |
| 2 | Technical Task | Implement data scraping | Build the scrapers and download data (JCDeaux & OpenWeather) |  |
| 3 | Technical Task | Frontend features | Design prototype for Web App with key features using Figma |  |
| 4 | Spike | Research Scrum Methodology | Understand the different aspects needed in Scrum documentation |  |
| 5 | Documentation Task | Report for project | Create a detailed report of ceremonies and artefacts |  |
| 6 | User Story |  |  |  |

***High Priority Features:***

1. **User story:** As a user, I want to sign up and create an account. -The system must provide a sign-up form with fields for name, email and password. -The email must be validated to ensure it is in the correct format. -The password must meet security requirements. -After successful sign-up, the user must receive a confirmation email with a link to verify their account.
2. As a user, I want log in so I can access my account. -The web app should provide a registration form with fields for name, email and password. -If the user entered a wrong name, email or password (incorrect credentials) and left any required field blank, an error message should be displayed. -After successful login, the user should be directed to the homepage of the system. -The user must be able to log out from their account at any time.

***Medium Priority Features:***

***Low Priority Features:***

Sprint Backlog:

|  |  |  |
| --- | --- | --- |
| **Item** | **Tasks** | **Status** |
| Software necessities & tools | Set up EC2, virtual environment, GitHub, Trello, Figma | Done |
| Implement data scraping | Build the scrapers from JCDecaux & OpenWeatherMap and download data | Done |
| Frontend features | Design prototype for Web App with key features using Figma | In Progress |
| Research Scrum Methodology | Understand the different aspects needed in Scrum documentation | Done |
| Report for project | Create a detailed report of ceremonies and artefacts for Sprint 1 | In Progress |