Appendices

[**Appendix A: Sprint Planning**](#_nhrgfoofxrgn) **30**

[**Sprint 00:**](#_s9p927s9ywlv) **30**

[**Sprint 01:**](#_m8jtkov4gaww) **31**

[**Sprint 02:**](#_1mm13isk30mz) **31**

[**Sprint 03:**](#_tyql3ovsn7du) **32**

[**Appendix B: Meeting Logs**](#_6kj9rqfkorha) **33**

[**Appendix C: Sprint Storyboards**](#_2tak5i4df8j3) **43**

[**Sprint 00**](#_fyoensa45lrr) **43**

[**Sprint 01**](#_c91nu08x1sm1) **45**

[**Sprint 02**](#_cp1eiel5ewhz) **48**

[**Sprint 03**](#_43bflbb78akg) **50**

[**Appendix D: Features & Product Backlog**](#_epyvg0iykl3u) **52**

# 

# 

# Appendix A: Sprint Planning

*Product Backlog = All features/products decided by team to complete our App (at the time)*

*Sprint Focus = Features/products team is focusing for each specific sprint*

## Sprint 00:

**Monday February 7th:**

Product Backlog:

* Scrum Process
* Project Management Tool
* Planning for structure of App (wireframe)
* SRS
* Issues with environments
* Scraper
* Scheduler
* Data Loader
* RDS
* Flask
* API for Bike Data
* API for Map
* API for Weather
* API for Weather Forecasting
* Display Available Bikes
* Display Available Stands
* Display Station Information
* Display Map of Dublin
* Display Weather Data
* Display locations of stations on map (markers)
* Display needed data in map markers of stations
* Predictive Model
* Display Predictions
* Display Prediction Chart(s)

Sprint Focus:

* Scrum Process
* Project Management Tool
* Planning for structure of App (wireframe)
* SRS
* RDS
* Scaper
* Scheduler
* API for Weather
* API for Bike Data
* Issues with environments

## Sprint 01:

**Monday February 21st:**

Product Backlog:

* Data Loader
* Flask
* API for Map
* API for Weather Forecasting
* Display Available Bikes
* Display Available Stands
* Display Station Information
* Display Map of Dublin
* Display Weather Data
* Display locations of stations on map (markers)
* Display needed data in map markers of stations
* Predictive Model
* Display Predictions
* Display Prediction Chart(s)
* Design plan for App
* Overall User features
* Create basic HTML/CSS layout
* Working with data from RDS

Sprint Focus:

* Flask
* Data Loader
* API for Map
* Display Map of Dublin
* Display locations of stations on map (markers)
* Create basic HTML/CSS layout
* Working with data from RDS
* Design plan for App
* Overall User features

## Sprint 02:

**Monday March 21st:**

Product Backlog:

* API for Weather Forecasting
* Display Available Bikes
* Display Available Stands
* Display Station Information
* Display Weather Data
* Display needed data in map markers of stations
* Predictive Model
* Display Predictions
* Display Prediction Chart(s)
* Data Grab/Handling
* Model Planning
* Initial Front-end features

Sprint Focus:

* API for Weather Forecasting
* Display Station Information
* Data Grab/Handling
* Model Planning
* Predictive Model
* Initial Front-end features

## Sprint 03:

**Monday April 4th:**

Product Backlog:

* Display Available Bikes
* Display Available Stands
* Display Weather Data
* Display needed data in map markers of stations
* Display Predictions
* Display Prediction Chart(s)
* Front-end improvements/finishings
* Finalise Model/Predictions
* Planning for report

Sprint Focus:

* Display Available Bikes
* Display Available Stands
* Display Weather Data
* Display needed data in map markers of stations
* Display Predictions
* Display Prediction Chart(s)
* Front-end improvements/finishings
* Finalise Model/Predictions
* Planning for report

# Appendix B: Meeting Logs

The Three Magic Questions:

“Yesterday, Today, Blockers”

1. What did you do yesterday?
2. What will you do today?
3. Anything blocking your progress?

Closing Questions: “Anyone looking for work?”, “Anything Else?”

**Week 1:**

2022/02/07 Mon, 11:30am:

**All:**

Informal Meeting, agree on use of “goodday.work” as PM tool

2022/02/08 Tue, 08:50:

**Tom:**

1. Planning, 2. Getting started with GitHub, 3. No.

**Will:**

1. Planning, 2. Getting started with GitHub, 3. No.

**Jörg:**

1. Planning, 2. Getting started with GitHub, 3. No.

2022/02/09 Wed, 08:50:

**Tom:**

1. Getting started with GitHub, 2. Starting out with AWS, RDS, etc., manage access issues, etc., 3. No.

**Will:**

1. Getting started with GitHub, 2. Starting out with AWS, RDS, etc., manage access issues, etc., 3. No.

**Jörg:**

1. Getting started with GitHub, 2. Starting out with AWS, RDS, etc., manage access issues, etc., 3. No.

2022/02/10 Thu, 08:50:

**Tom:**

1. Planning/Code Management/Scheduler, 2. Planning/Code Management/Scheduler, 3. No.

**Will:**

1. Planning/Documentation/Database Design, 2. Planning/Documentation/Database Design, 3. No.

**Jörg:**

1. Planning/Documentation/SRS, 2. Planning/Documentation/SRS, 3. No.

2022/02/11 Fri, 11:15:

**Tom:**

1. Planning/Code Management/Scheduler, 2. Planning/Code Management/Scheduler, 3. No.

**Will:**

1. Planning/Documentation/Database Design, 2. Planning/Documentation/Database Design, 3. No.

**Jörg:**

1. Planning/Documentation/SRS, 2. Planning/Documentation/SRS, 3. No.

**Week 2:**

2022/02/14 Mon, 12:45 (Walked the board):

**Tom:**

1. Github documentation, 2. Data Loader, 3. No.

**Will:**

1. Database Implementation, 2. Add sql schema to git, 3. No.

**Jörg:**

1. Planning/Documentation/SRS, 2. Planning/Documentation/SRS, 3. No.

2022/02/15 Tue, 12:15 (Round Robin):

**Tom:**

1. Started the data loader, 2. I plan to resolve the db connection issue with the loader, 3. No.

**Will:**

1. Created the station table/database, backup freq changed, 2. Git / Virtual env’s, backup instructions, 3. No.

**Jörg:**

1. New User stories, SRS, wireframe in SRS, file to drive 2. Git / Virtual env’s and resolve issue with virtual environment on mac , 3.No

2022/02/16 Wed, 15:45 (Round Robin):

**Tom:**

1., 2., 3. No.

**Will:**

1. Created JSON ImportLog Table, 2.Exported SQL Schema/Shared RDS Info, 3. Using right branch for SQL Schema upload (will working on tomorrow

**Jörg:**

1. Still trying to resolve the incompatibility issues with the virtual environment export/import (yml/txt) across different platforms (MacOS and Windows).

2022/02/17 Thurs, (Round Robin):

**Tom:**

1.Database RDS look/DataLoader(WIP)/Eniv Issues, 2.Word Doc of Installed/Database Talk/Weather Talk/ML Talk/Update Database, 3. No.

**Will:**

1.Exported SQL Schema/Shared RDS Info, 2.Database Talk/Weather Talk/ML Talk/Sample Weather Data, 3. No.

**Jörg:**

1.Still trying to resolve the incompatibility issues with the virtual environment export/import (yml/txt) across different platforms (MacOS and Windows). , 2. , 3.No.

2022/02/18 Fri, (Round Robin):

**Tom:**

1.Word Doc of Installed/Database Talk/Weather Talk/ML Talk/Update Database, 2.Installer Merge with Data Loader into Scheduler, 3. No.

**Will:**

1.Database Talk/Weather Talk/ML Talk/Sample Weather Data/Goodday Research/Weather Sample Data/Committed Sample Weather Data, 2. Exported and Committed New Version of Sql Schema/Goodday Research, 3. No.

**Jörg:**

1.Dataloader Planning/Dataloader, 2. Implement get station-ID into data loader 3.No.

**Week 3:**

2022/02/21 Mon, (Round Robin):

**Tom:**

1. Sprint Retro & Review, Create weather storage table (key), add timestamp to stationState, get weather data from API, stored data in database, merge all branches (and removed them), installed latest to EC2, tested the scheduler, fix issues with virtual environment, , 2. Sprint 1 planning, 3. No.

**Will:**

1. Sprint Retro & Review/Goodday Research & Communication, 2.Goodday Research/Prepare Next Sprint Scrum Techniques, 3. No.

**Jörg:**

1.Sprint Retro & Review, Get station-id integrated into dataloader.py 2. Update datetime format in RDS, 3.No.

2022/02/22 Tue, (Round Robin):

**Tom:**

1. Fixed bug with lastUpdate timestamp and location weather, 2. Sprint Planning, 3. Project Management tool because we’re facing issues with the goodday burndown chart.

**Will:**

1.Project Management Tool Research/Scrum Planning, 2.Planning/Project Managment Tool, 3. Yes: Bugs/Research of Project Management Tool.

**Jörg:**

1. Datetimeformat updated in RDS, 2. Sprint Planning/, 3. No.

2022/02/23 Wed, (Round Robin):

**Tom:**

1. Sprint Planning/Installer/Flask, 2. Sprint Planning - yes - still on Wednesday, even more discussion on our wireframes. , 3. No.

**Will:**

1.Project Management Tool Research/Sprint Planning/Installed Installer, 2.Sprint Planning, 3., No.

**Jörg:**

1. Creating and adding user stories to goodday, 2. Sprint Planning, Gathering ideas to draft first wireframe /, 3. No.

2022/02/24 Thu , (Round Robin):

**Tom:**

1. Bash script to manage scheduler. Complete but needs to be installed., 2. Updating the Setup.py to create an endpoint for the management script. and testing., 3. No.

**Will:**

1.Sprint Planning, 2.Sprint Planning/Installing, 3., No.

**Jörg:**

1. Collecting ideas to draft the first wireframe, 2. Sprint Planning, drawing first wireframe, 3. No.

2022/02/25 Fri, (Round Robin):

**Tom:**

1. Installing the scheduler management script, 2. Sorting out the issue with our virtual environment not being available to root when we run sudo, 3. No.

**Will:**

1.Sprint Planning, 2.Goodday Research/Burndown options, 3. Yes: Goodday Burndown capabilities.

**Jörg:**

1. Drawing first wireframe/DB Copy, 2. Implementing first basic HTML/CSS layout, 3. No.

**Week 4:**

2022/02/28 Mon , (Round Robin):

**Tom:**

1. DB Copy, getting Setup.py to create an endpoint for our scheduler script. 2. Looking into web servers for the EC2 Instance, 3. No.

**Will:**

1.Goodday Research/Burndown options, 2.Sprint Burndown Template Creation/EC2 Costs, 3., No.

**Jörg:**

1. Implementing first basic HTML/CSS layout, 2. Still ongoing - implementing first basic HTML/CSS layout, 3. No.

2022/03/01 Tue, (Round Robin):

**Tom:**

1. Flask Server (Apache Server), 2. Flask Server (Apache Server), 3. Maybe.

**Will:**

1.Sprint Burndown Template Creation/EC2 Costs, 2.Sprint 0 Burndown/Google Maps API, 3., No.

**Jörg:**

1. Implementing first basic HTML/CSS layout, 2. Enhancing first basic HTML/CSS layout , 3. No.

2022/03/02 Wed, (Round Robin):

**Tom:**

1. Apache server - investigating issues with WSGI, 2. Apache server - investigating issues with WSGI (Mod\_WSGI compiled with wrong python), 3. Yes

**Will:**

1. Sprint 0 Burndown/Google Maps API, 2.Google Maps HTML/CSS/JS code structures, 3., No.

**Jörg:**

1. Enhancing first basic HTML/CSS layout, 2. Further enhancing first basic HTML/CSS layout, 3. No.

2022/03/03 Thu, (Round Robin):

**Tom:**

1. Apache server - investigating issues with WSGI (Mod\_WSGI compiled with wrong python), 2. Finally resolved the issues - our python (flask) application is live! Woots!, 3. No.

**Will:**

1. Google Maps HTML/CSS/JS code structures, 2. Embedding Map and Markers for Web App, 3., No.

**Jörg:**

1.Further enhancing first basic HTML/CSS layout, 2. Still ongoing - further enhancing first basic HTML/CSS layout, 3. No.

**Week 5:**

2022/03/21 Mon, (Round Robin):

**Tom:**

1.Integrating maps, styles, basic html, javascript, etc. 2. Update the release (setup.py) file, merge dev into main, and then do an install 3. No

**Will:**

1.Embedding Map and Markers for Web App 2. Sprint planning, Collecting ideas how to implement SQL query to get current weather/occupancy displayed on the front end 3. No

**Jörg:**

1.Further enhancing first basic HTML/CSS layout2. Sprint planning, Collecting ideas how to implement SQL query to get current weather/occupancy displayed on the front end 3. No

2022/03/22 Tues, (Round Robin):

**Tom:**

1. Really just admin, do a release to the website, 2. <Tom out sick today>, 3. No

**Will:**

1.Sprint planning, Collecting ideas how to implement SQL query to get current weather/occupancy displayed on the front end, 2. , 3. No

**Jörg:**

1.Sprint planning, Collecting ideas how to implement SQL query to get current weather/occupancy displayed on the front end, 2. , 3. No

2022/03/23 Wed, (Round Robin):

**Tom:**

1. <Tom out sick today>, 2. <Tom out sick today>, 3. No

**Will:**

1. , 2. , 3. No

**Jörg:**

1. , 2. , 3. No

2022/03/24 Thu, (Round Robin):

**Tom:**

1. <Tom out sick today>, 2. Add basic tables for resampled data to the database to support Jörg and Will, 3. No

**Will:**

1. , 2. Starting to design a basic model for ML , 3. No

**Jörg:**

1. , 2. Thinking about the data input for the machine learning model, 3. No

2022/03/25 Fri, (Round Robin):

**Tom:**

1. Just basic admin - add tables to database(s) for ML model, 2. Looking at sprint review and sprint retrospective., 3. No

**Will:**

1. Starting to design a basic model for ML, 2. Implementing a basic version of the ML model , 3. No

**Jörg:**

1. Thinking about the data input for the machine learning model, 2. Starting to implement the resample collected data (occupancy, weather) into a more suitable format for the ML model, 3. No

**Week 6:**

2022/03/28 Mon, (Round Robin):

**Tom:**

1. Added colored icons for the bikes on the map, 2. Adding an end-point for the occupancy data , 3. No

**Will:**

1. DataPrep/Created a basic model for ML, 2. Sprint planning , 3. No

**Jörg:**

1. Having started resampling the weather/occupancy data, 2. Continue to resample the weather/occupancy data, 3. No

2022/03/29 Tues, (Round Robin):

**Tom:**

1. Small server tweaks, adding the end-point for the occupancy data, 2. Including the doesn’t-yet-exist data\_resampler into the scheduler and installer, 3. No

**Will:**

1. Sprint Planning, 2. Created Pickle from Model/Deserialized Pickle for dwmb.py, 3. No

**Jörg:**

1. Sprint Planning, 2. Querying occupancy data from the RDS using pandas , 3. External factors have prevented me from progressing: The urge to install JavaFx and set up project collaboration using Eclipse/Github to get up and running with the Java group project.

2022/03/30 Wed, (Round Robin):

**Tom:**

1.Integrating the data\_resampler into the scheduler and installer, 2. Tidying up, refactoring scheduler , 3. No

**Will:**

1. Created Pickle from Model/Deserialized Pickle for dwmb.py, 2. Logistic Model, 3. No

**Jörg:**

1.Querying occupancy data from the RDS using pandas , 2. Start implementing data resampler in Jupyter Notebook, 3. External factors have prevented me from progressing: The urge to install JavaFx and set up project collaboration using Eclipse/Github to get up and running with the Java group project.

2022/03/31 Thur, (Round Robin):

**Tom:**

1. Tidying up, refactoring scheduler, 2., 3. No

**Will:**

1. Logistic Model, 2. Logistic Model, 3. No

**Jörg:**

1. Started implementing data resampler in Jupyter Notebook, 2. Implementing function “station state hourly” - data resampler, 3. No

2022/04/01 Fri, (Round Robin):

**Tom:**

1., 2. Front-end slider work, 3. No

**Will:**

1. Logistic Model, 2. Predict Route/Forecast Weather API/Working with Forecast Weather Return, 3. No

**Jörg:**

1. Implementing function “resample station state hourly” - data resampler, 2. Testing function “resample station state hourly” - data resampler, 3. No

**Week 7:**

2022/04/04 Mon, (Round Robin):

**Tom:**

1., 2. Fix google maps API Key for live, release version live (includes resampler), 3. No

**Will:**

1. Predict Route/Forecast Weather API/Working with Forecast Weather Return, 2. Sprint Planning/Sprint Review, 3. Deadline moved up two days per Prof. Lawlor

**Jörg:**

1. Testing function “resample station state hourly” - data resampler, 2. Implementing function “resample weather state hourly” - data resampler, 3. No

2022/04/05 Tue, (Round Robin):

**Tom:**

1.Release, Maps, 2. Fix bug in resampler, rename ‘dl\_loader’ to ‘scheduler’ and re-install, 3. No

**Will:**

1. Sprint Planning/Sprint Review, 2. Prototyped the results for the Predict Route, 3. No

**Jörg:**

1. Implementing function “resample weather state hourly” - data resampler, 2.Keep on implementing function “resample weather state hourly” - data resampler, 3. No

2022/04/06 Wed, (Round Robin):

**Tom:**

1. Fix bug, rename scheduler, 2. Work on using ORM in python back end for data loading (basic queries, joins, filters (aka where clauses), 3. No

**Will:**

1. Prototyped the results for the Predict Route, 2. Updated and/or created Multi Linear Reg & Random Forest Models/Create Skeleton for Project Report, 3. No

**Jörg:**

1 .Keep on implementing the function “resample weather state hourly” - data resampler, 2. Testing function “resample weather state hourly” - data resampler, 3. No

2022/04/07 Thu, (Round Robin):

**Tom:**

1. SQL Alchemy ORM, 2. flask endpoint for prediction model, flask endpoint for occupancy, 3. No

**Will:**

1. Updated and/or created Multi Linear Reg & Random Forest Models, 2. Evaluated Models/Worked on pickle files and their size issues, 3. No

**Jörg:**

1.Testing function “resample weather state hourly” - data resampler, 2. Refactoring prototyping code of data resampler from Jupyter Notebook into clean Python code , 3. No

2022/04/08 Fri, (Round Robin):

**Tom:**

1. Flask endpoints, 2. Worked with Will on splitting the prediction model from single ‘monolith’ to model per station, focusing on reduced size to reduce load times from current 55s to circa 5, 3. No

**Will:**

1. Evaluated Models/Worked on pickle files and their size issues, 2. Worked on modifying pickle file to lower the load of running it on the app with Tomas, 3. No

**Jörg:**

1.Refactoring prototyping code of data resampler from Jupyter Notebook into clean Python code, 2. Implementing search mode ‘available bikes’ & ‘available spaces’ in Javascript front-end, 3. No

**Week 8:**

2022/04/11 Mon, (Round Robin):

**Tom:**

1. gviz api - introduced and done (for occupancy charts), complete server rebuild (from scratch) on new EC2 instance, fix map markers not zooming and not displaying correctly , 2. <nothing! - mad java push today>, 3. No

**Will:**

1. Worked on modifying pickle file to lower the load of running it on the app with Tomas, 2. Grabbed weather icons for weather display on app/Created logic and function to dynamically display weather icons due to weather forecasts, 3. No

**Jörg:**

1.Implementing search mode ‘available bikes’ & ‘available spaces’ in Javascript front-end, 2. Start implementing ‘getBikeIconUrl’ in Javascript front-end, 3. No

2022/04/12 Tue, (Round Robin):

**Tom:**

1. Install hotfix live - was a couple of days overdue. Issues remain, but closer, 2. Project report, 3. No

**Will:**

1. Grabbed weather icons for weather display on app/Created logic and function to dynamically display weather icons due to weather forecasts, 2. Writing Data Analytics & Process for Report , 3. No

**Jörg:**

1.Start implementing ‘getBikeIconUrl’ in Javascript front-end, 2. Completing the implementation of ‘getBikeIconUrl’ in Javascript front-end. Creating the function ‘createMarkers’ in the front-End, 3. No

2022/04/13 Wed, (Round Robin):

**Tom:**

1. Dealing with outstanding issues, 2. Project report (Architecture section), 3. No

**Will:**

1. Writing Data Analytics & Process for Report, 2. Continued writing and editing for report, 3. No

**Jörg:**

1.Completing the implementation of ‘getBikeIconUrl’ in Javascript front-end. Creating the function ‘createMarkers’ in the front-End,, 2 Implementing range slider for date&time selection in Javascript front-end., 3. No

2022/04/14 Thu, (Round Robin):

**Tom:**

1. Project report, 2. Update front end to display predicted data on the map when slider changes (was not triggering), 3. No

**Will:**

1. Continued writing and editing for report, 2. Updated some model details/Finishing first draft of data analytics and process for report/Report editing, 3. No

**Jörg:**

1.Implementing range slider for date&time selection in Javascript front-end, 2. Bug Fixing Javascript front-end, 3. No

2022/04/15 Fri, (Round Robin):

**Tom:**

1. Getting predicted data to display properly on map, 2.Final updates to the front end, add map legend, remove unnecessary data loads , 3. No

**Will:**

1. Updated some model details/Finishing first draft of data analytics and process for report/Report editing, 2. Final Burndown/Writing for reflection and meeting log section of report/Title page, table of contents, appendix, formatting, and final editing/PDF report & Submit report, 3. No

**Jörg:**

1.Cleaning up dead code from Javascript front-end, 2. DONE!, 3. No.

# Appendix C: Sprint Storyboards

## Sprint 00

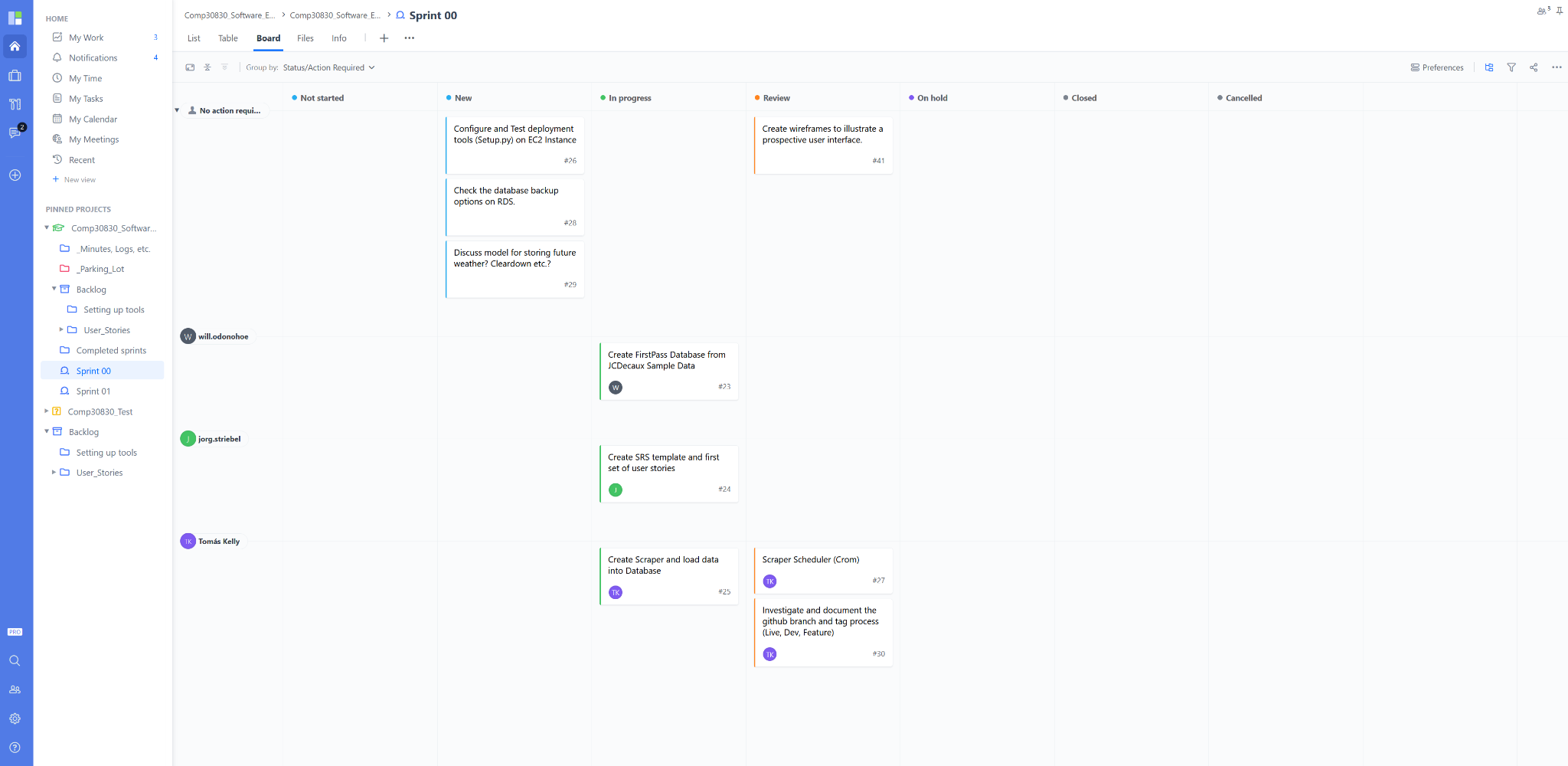
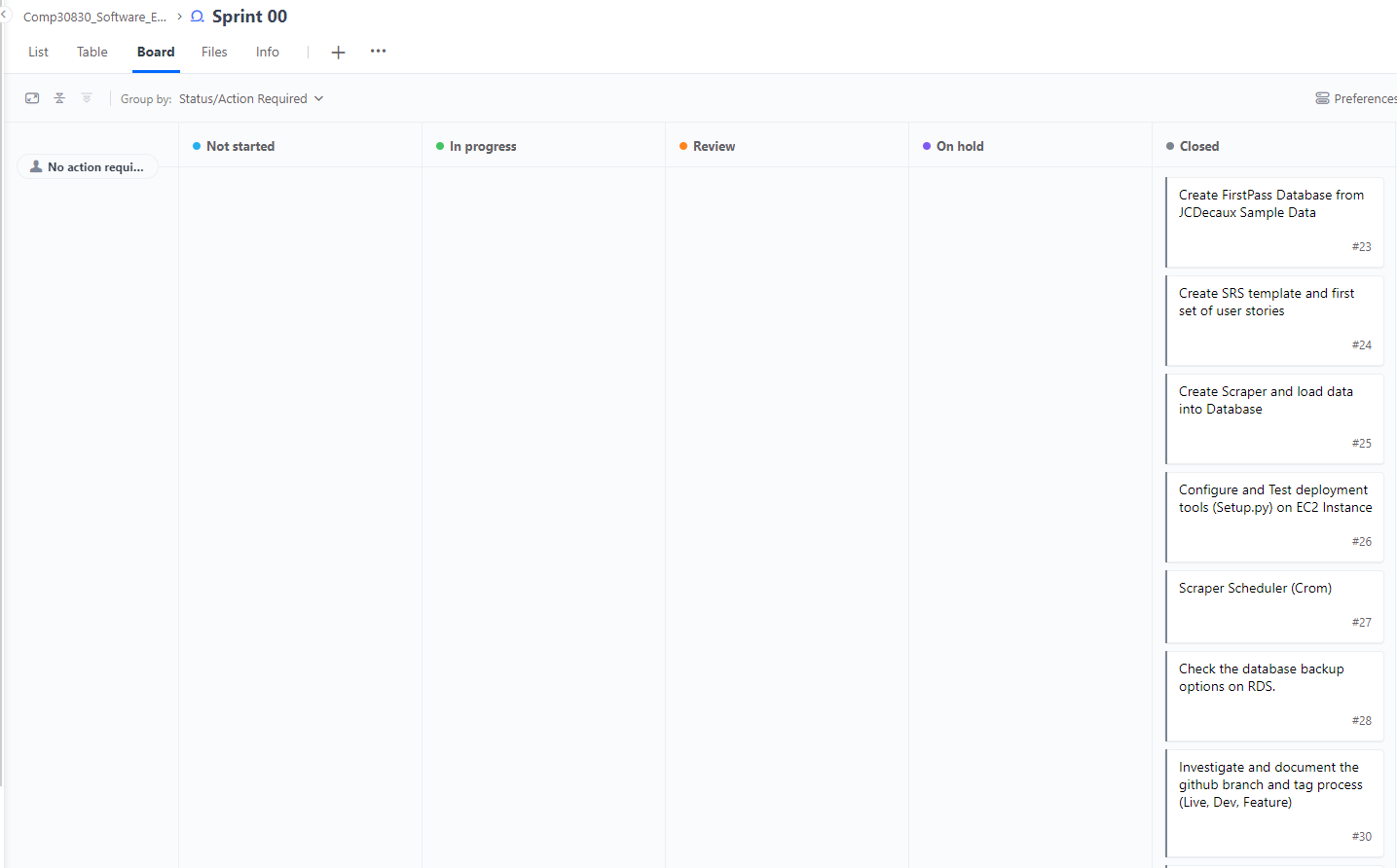


Figure 18: End of the first week of Sprint 00 - Storyboard



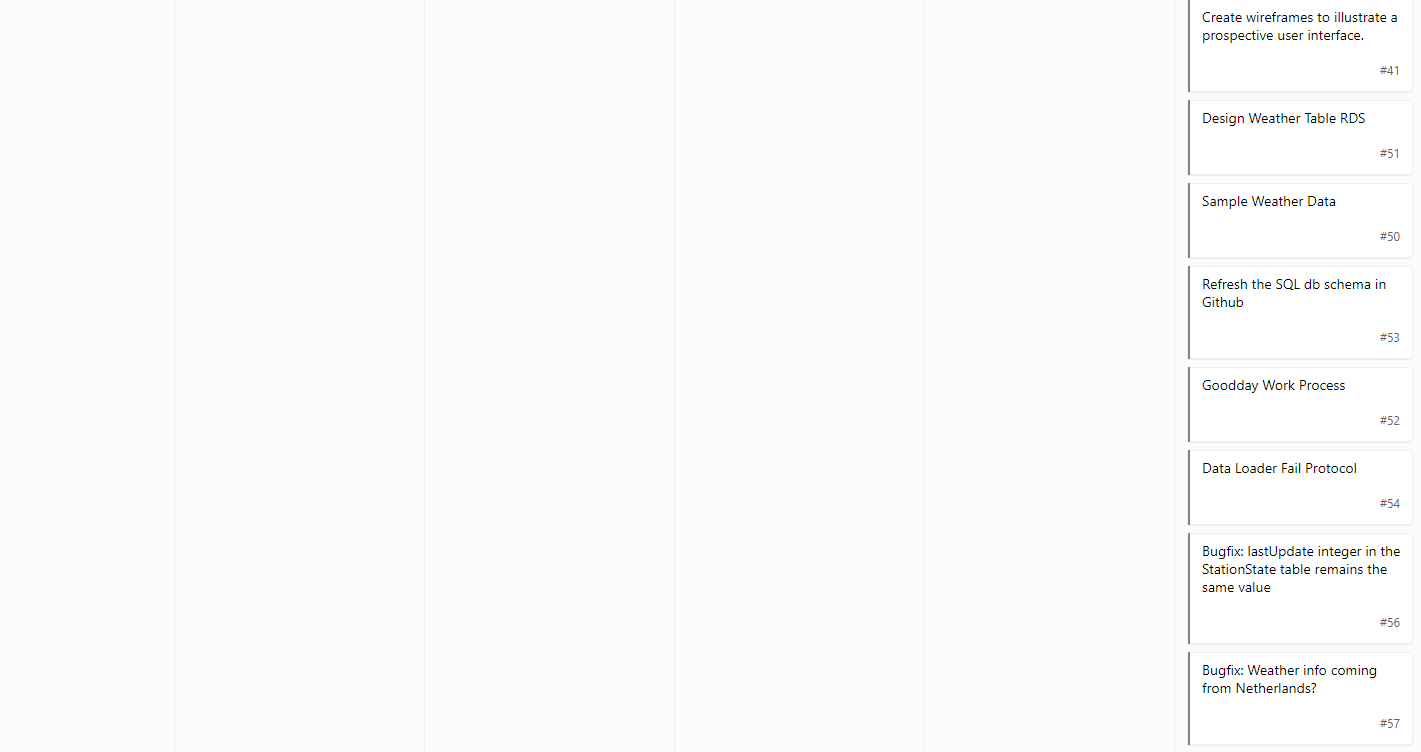


Figure 19: End of Sprint 00 - Storyboard

## 

## Sprint 01

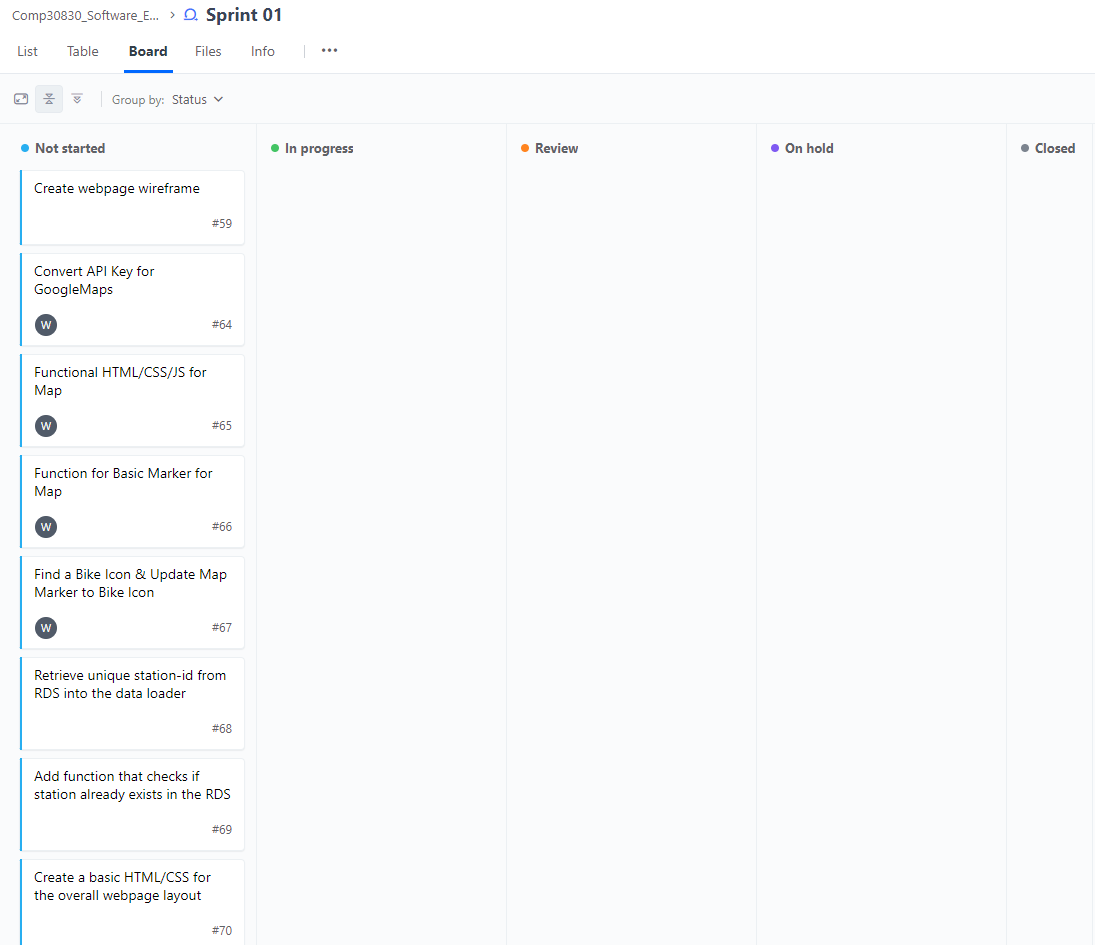




Figure 20: Start of Sprint 01 - Storyboard

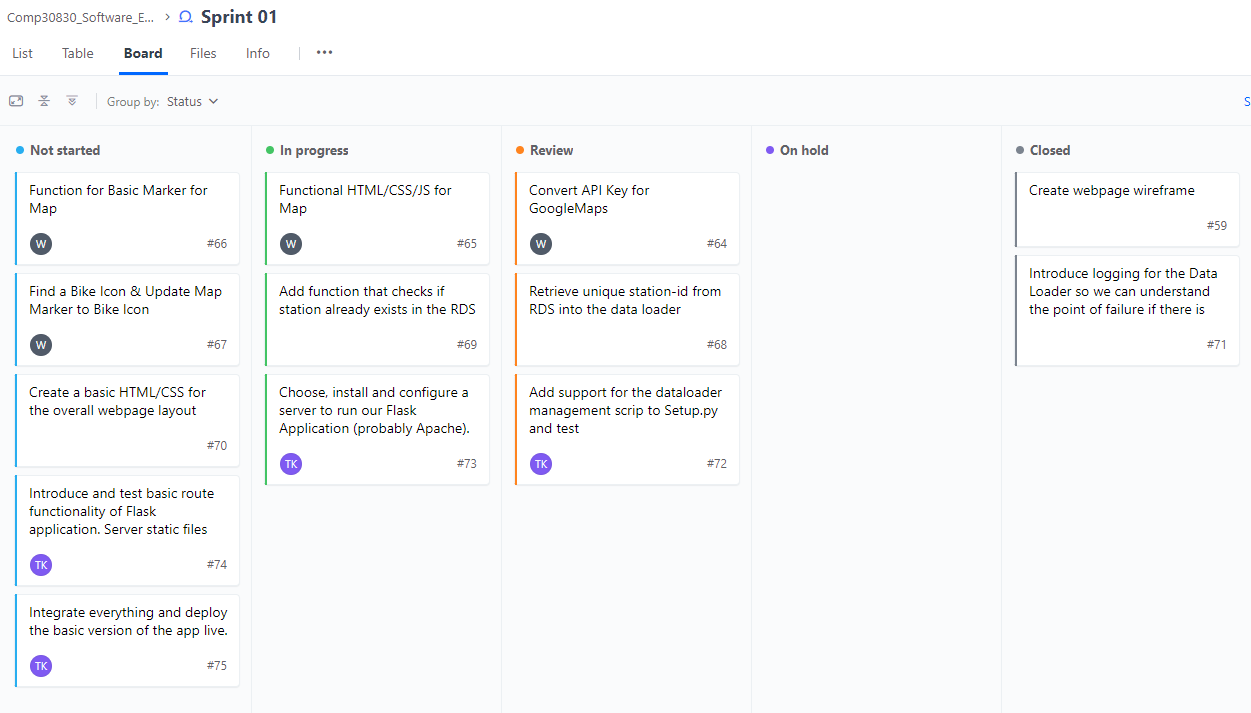
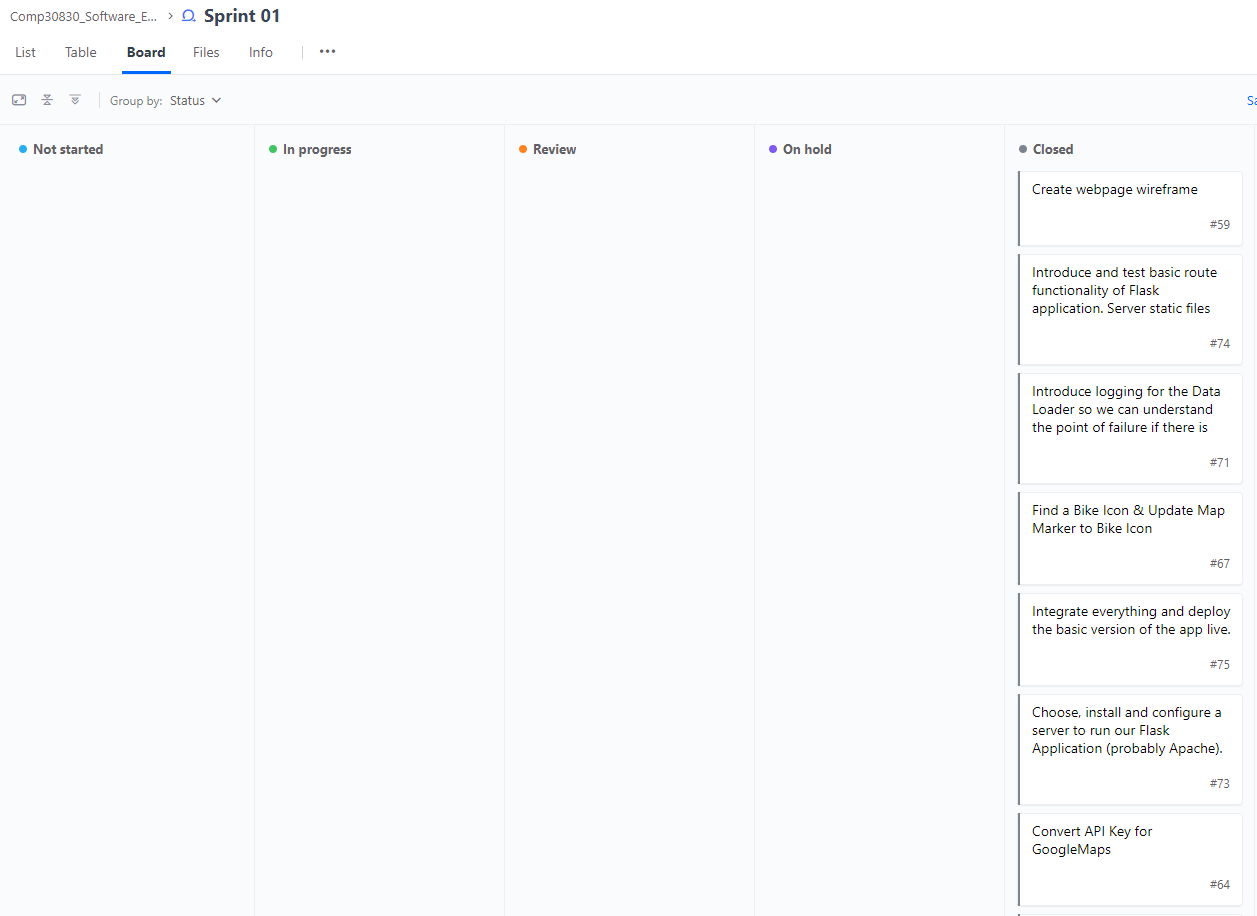


Figure 21: End of the first week of Sprint 01 - Storyboard



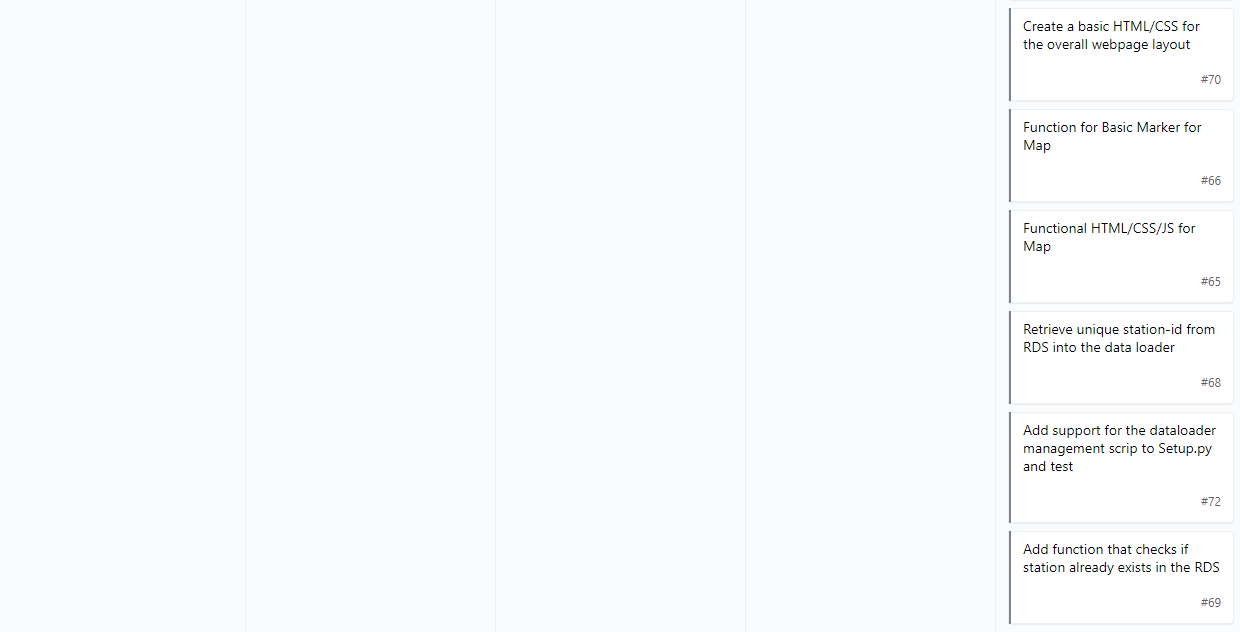
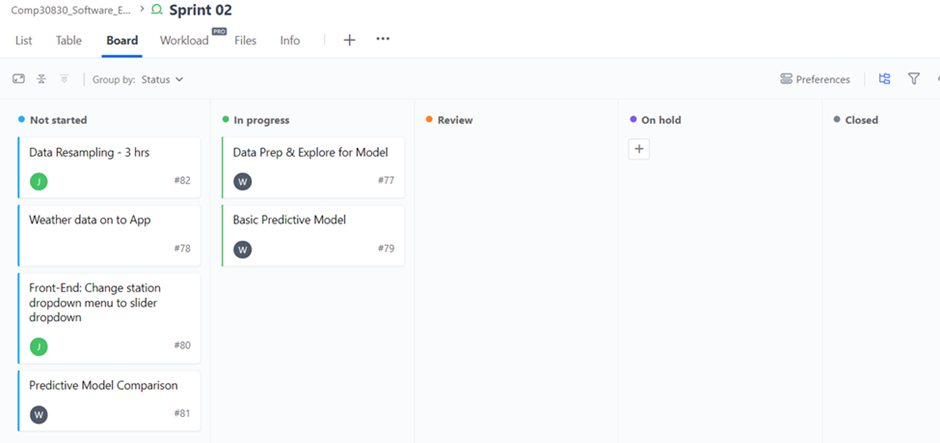
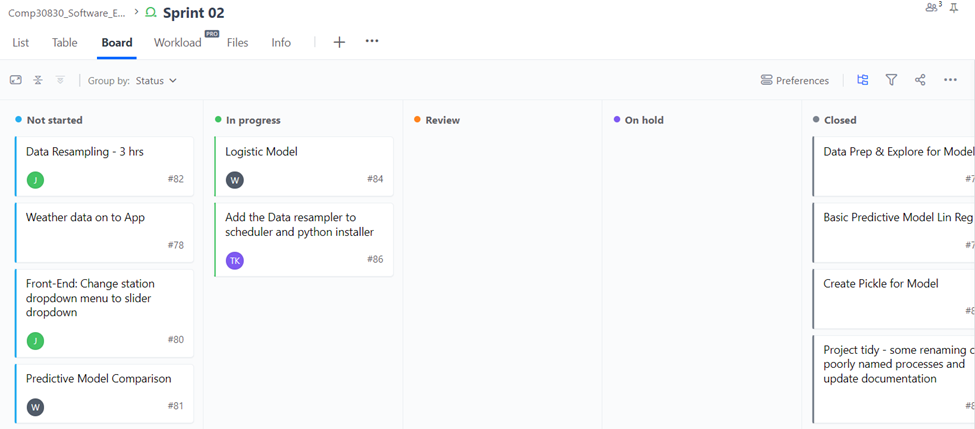


Figure 22: End of Sprint 01 - Storyboard

## 

## Sprint 02

****Figure 23: Start of Sprint 02 - Storyboard

Figure 24: End of the first week of Sprint 02 - Storyboard

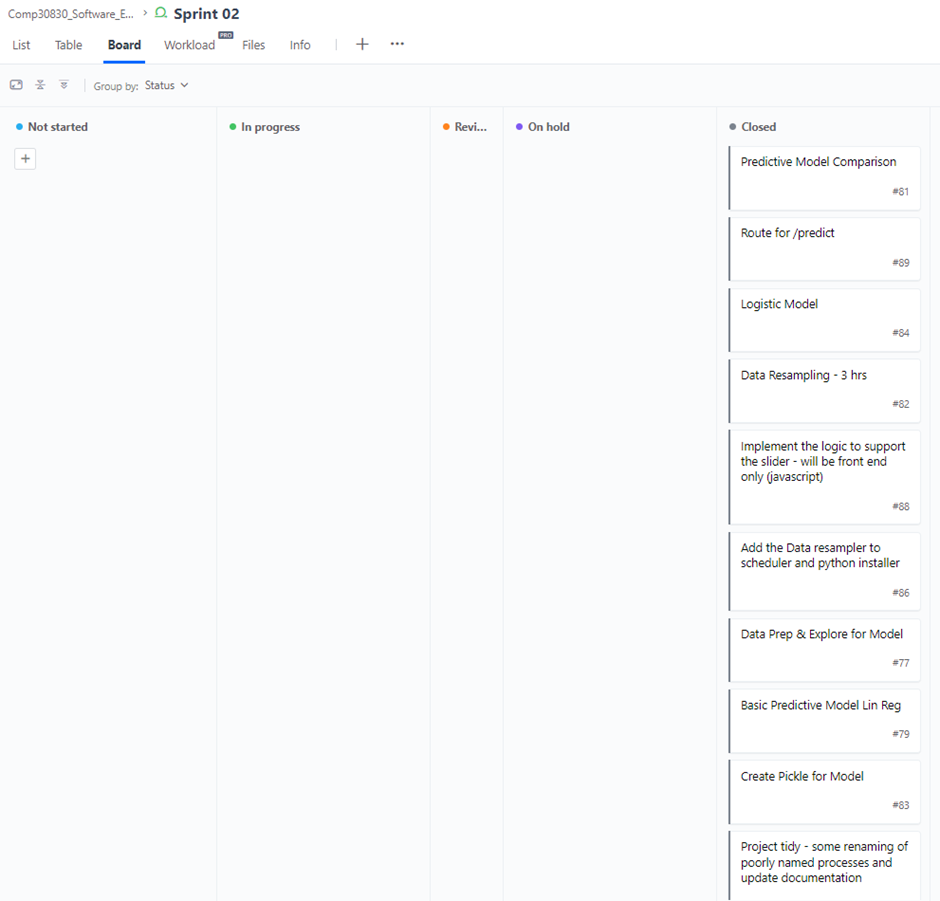
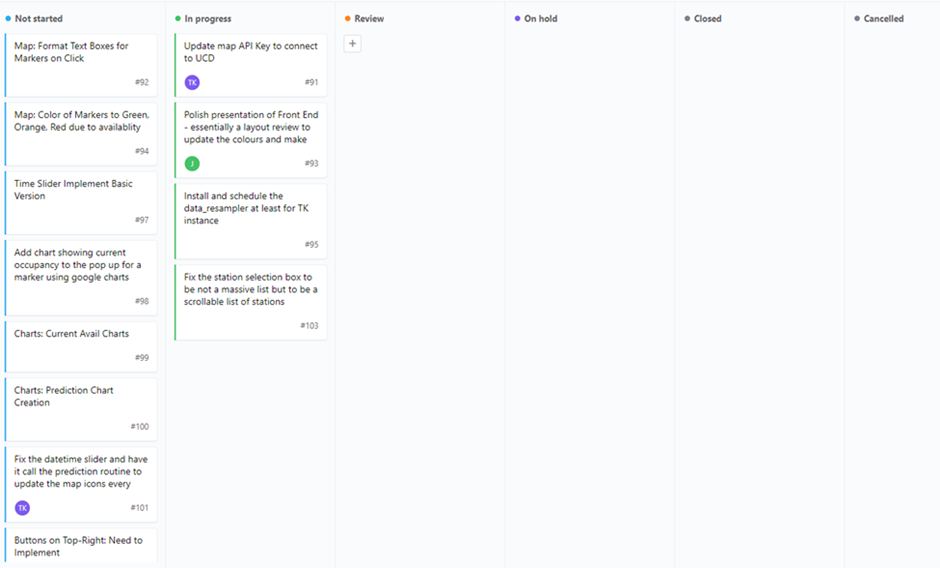
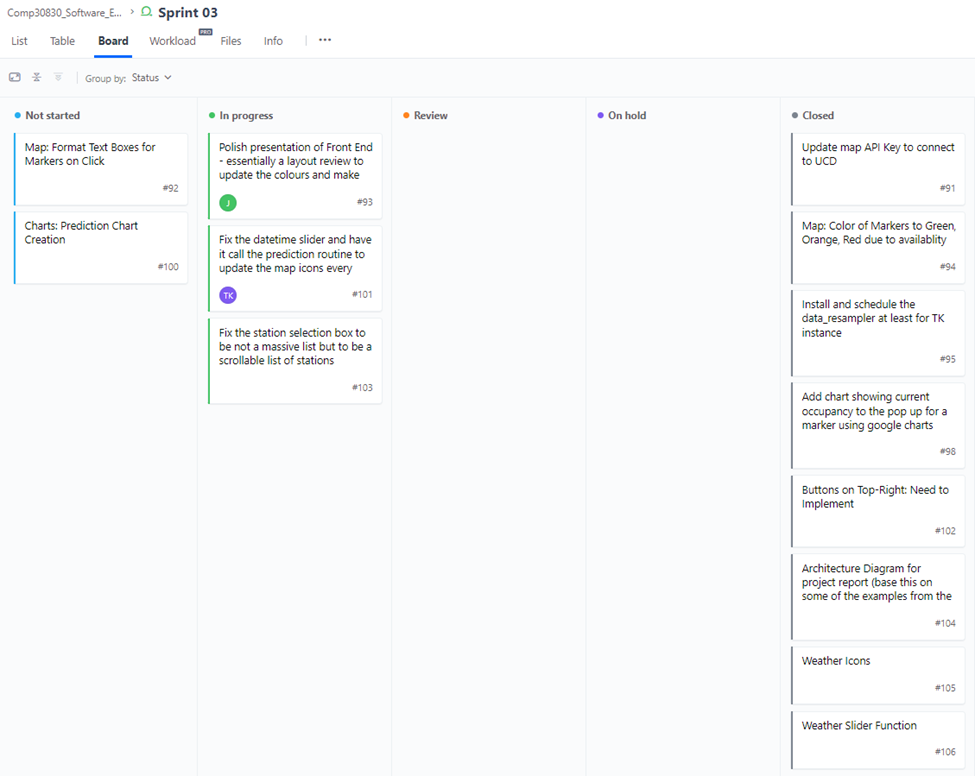


Figure 25: End of Sprint 02 - Storyboard

## 

## Sprint 03

Figure 26: Start of Sprint 03 - Storyboard

Figure 27: End of the first week of Sprint 03 - Storyboard

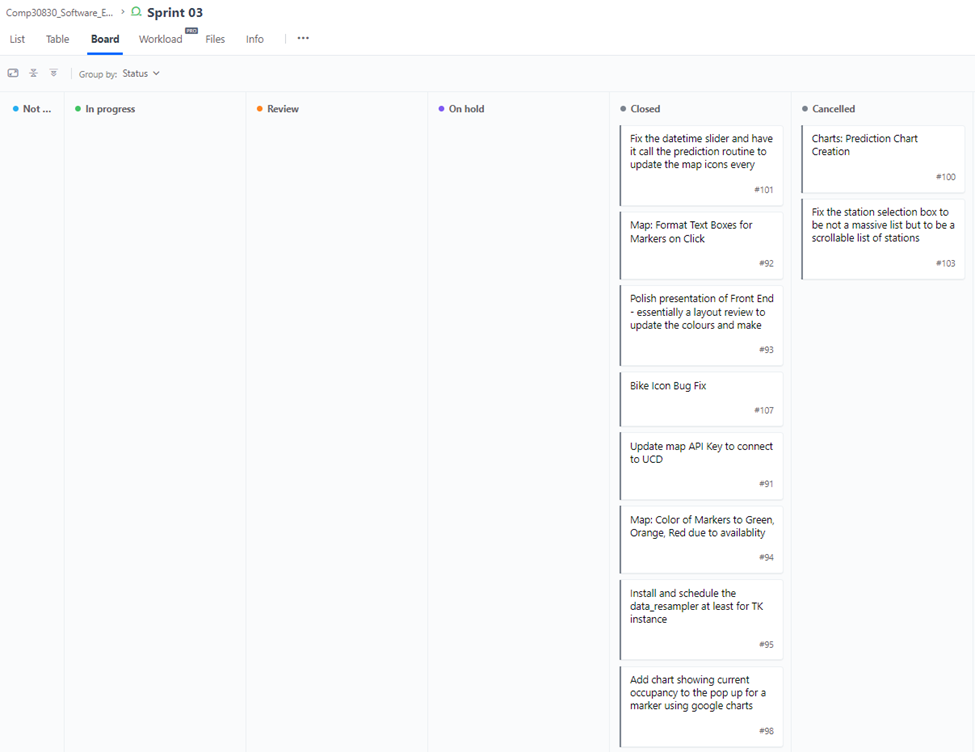




Figure 28: End of Sprint 03 - Storyboard

# Appendix D: Features & Product Backlog

An “X” indicates in which sprint the backlog item was addressed.

Table 4: Feature & Product Backlog Across All Sprints

| **Feature** | **Description** | **Product Backlog** | **Sprint 00** | **Sprint 01** | **Sprint 02** | **Sprint 03** |
| --- | --- | --- | --- | --- | --- | --- |
| **Overall Function** | The back-end and structures that make the application run. | Wireframe structure of App | **X** |  |  |  |
| Scraper | **X** |  |  |  |
| Scheduler | **X** |  |  |  |
| Data Loader |  | **X** |  |  |
| RDS | **X** |  |  |  |
| Flask |  | **X** |  |  |
| Design App |  | **X** |  |  |
| Create HTML/JS/CSS Structure |  | **X** |  |  |
| RDS Database Structure/Implementation | **X** |  |  |  |
| **Map** | An interactive map where the user can see where bike stations are located in Dublin. The station markers will have station information | Google Maps API |  | **X** |  |  |
| JCDecaux API for bike data | **X** |  |  |  |
| Display Dublin on Map |  | **X** |  |  |
| Display markers for station location |  | **X** |  |  |
| Display Available Bikes |  |  | **X** |  |
| Display Available Spaces |  |  | **X** |  |
| Display Station Information |  |  | **X** |  |
| Display information in map markers |  |  | **X** |  |
| **Available Bikes/Spaces Buttons** | Two buttons where the user can choose between bikes or spaces depending if they are looking for a bike or looking to return a bike. | Functionality for Available Bikes |  |  |  | **X** |
| Functionality for Available Spaces |  |  |  | **X** |
| **Station Dropdown** | An interactive dropdown where the user can choose which station to focus on. Will update necessary data/information. | Create Dropdown |  | **X** |  |  |
| Grab Station Information | **X** |  |  |  |
| Use Station Name for Dropdown |  | **X** |  |  |
| **Date Slider** | An interactive slider for the user to choose to focus on a time in the future. Will update necessary data/information/charts. | Create Slider |  |  | **X** |  |
| Open Weather API |  |  | **X** |  |
| Open Weather Forecast (one call) API |  |  | **X** |  |
| Connect data to slider |  |  |  | **X** |
| **Past Availability Chart** | A chart that shows the user past bike data | Grab data from RDS |  | **X** |  |  |
| Display bike data |  |  |  | **X** |
| Create past availability chart |  |  |  | **X** |
| **Future Availability Chart** | A chart that shows the user future predictions of bike/space availability based over past bike and weather data, and weather forecasts. | Data Handling |  |  | **X** |  |
| Data Cleaning |  |  | **X** |  |
| Model Planning |  |  | **X** |  |
| Model Comparisons |  |  | **X** |  |
| Integrating Model into App |  |  |  | **X** |
| Output Predictions into chart |  |  |  | **X** |