L-Università ta' Malta
Faculty of Information & Communication Technology
Department of Computer Information Systems

Group Assigned Practical Task

Kyle Agius* (357302L), Neil Bugeja** (51000L), Isaac Borg** (133801L), Andrea Fenech Cesareo** (74802L)

*B.Sc. (Hons) Software Development

**B.Sc. (Hons) Computing and Business

Study-unit: **Group APT**

Code: **CIS2107**

Lecturers: **Christopher Porter, Ernest Cachia**

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Declaration

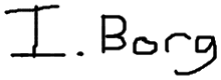
Plagiarism is defined as “the unacknowledged use, as one's own, of work of another person, whether or not such work has been published, and as may be further elaborated in Faculty or University guidelines" (University Assessment Regulations, 2009, Regulation 39 (b)(i), University of Malta).

We, the undersigned, declare that the assignment submitted is our work, except where acknowledged and referenced.

We understand that the penalties for committing a breach of the regulations include loss of marks; cancellation of examination results; enforced suspension of studies; or expulsion from the degree programme.

Work submitted without this signed declaration will not be corrected and will be given zero marks.

Kyle Agius _____ _____
Student Name Signature



Isaac Borg _____ _____
Student Name Signature

Neil Bugeja _____ _____
Student Name Signature

Andrea Fenech Cesareo _____ _____
Student Name Signature

CIS2107 _____ Group Assigned Practical Task ___
Course Code Title of work submitted

24/05/2022 _____
Date

Table of Contents

1 – Introduction

Flight simulation is a popular activity which involves accurately recreating the experience of piloting a plane on a computer. This project aims to tackle issues faced when planning and running flights on a simulator. The following subsections shall introduce the basics of flight simulation, as well as what we, as team members, aim to accomplish and how we shall divide our efforts.

1.1 – General Domain

The project is focused on the flight simulation domain. This involves people using their home computer in order to simulate a real-life flight with a significant attention to realism. There are numerous flight simulation programs one can opt to use, the most prominent being Microsoft Flight Simulator 2020. It is so prominent, in fact, that a recent survey [1] found that 75% of simmers use this program. While this can be used with a standard keyboard and mouse, some simmers make use of additional hardware, such as a yoke, to have a more authentic experience. In addition to being a hobby, flight simulation is also used in real-word pilot training. This assists students in understanding piloting fundamentals in a safe environment.

In an initial meeting with our lecturer [2], we were introduced to some of the general concepts regarding flight simulation. This included a brief introduction of the flight process and a look at various third-party software that can be used to plan flights.

Notably, not everyone sims in the same way. Some simmers pay very close attention to realism, while others are more casual. This can be seen by how closely they follow standard procedures. In addition, flights can follow Visual Flight Rules (VFR) or Instrument Flight Rules (IFR). As such, any flight planning and monitoring program should be capable of supporting the needs of various simmers.

For the flight to be considered to follow standard procedures, the different phases of flight can be considered to be the steps needed. For this project, we are considering nine main phases of flight [3]. The first is the Pre-flight, which involves performing various checks and planning before the plane starts moving. The plane is then taxied to the designated runway, which is, in itself, a phase. Next, the Take-off phase consists of when the plane lifts from the ground, and the Climb phase consists of the ascent of the plane to the predetermined cruising altitude. Once the plane reaches this altitude, the flight is considered to be in the Cruise phase, which is generally the longest in the flight and the least demanding on the pilot. Once the plane starts getting closer to its destination, the plane starts the process of lowering in altitude to eventually land. This start is known as the Descent phase. Next, the Approach phase is when the plane prepares for landing and the Landing phase, a particularly challenging task for pilots, is when the plane touches ground. Finally, an additional Taxi phase is required at the destination airport.

1.2 – Task Definition

The aim of the project is to create a prototype web application which flight simmers would be able to use when planning their flight. The web app should be able to provide all the required information that a simmer may need, and can be used alongside the flight so that all data can be accessed easily when needed.

User experience should be considered so that the app is user-friendly. This includes limiting information overload and displaying information in a clear and visually appealing manner. The application should favour showing information through visual means rather than having large quantities of text, with a good hierarchy and consistency. This compresses the space needed for the data and makes it faster and easier to pinpoint the particular information needed at any moment.

Interaction with APIs is very important for such a project, as the program would need to gather various data points from different sources in order to display all the information needed, as well as ensuring correctness of the data.

Since the project is primarily about the UI, a detailed design should be made before the creation of the application.

1.3 – Splitting of Work

1.3.1 – Research and Design

- Joint
    - General design
- Andrea
    - Pre-flight
    - Figma Designs
- Neil:
    - Simulators and communication, colour schemes
    - Colour Schemes
    - Design of notepad and account feature
- Isaac:
    - Taxi, Take-off, Landing
    - Use Case Diagram
- Kyle:
    - Climb, Cruise, Descent and Approach.
    - Early concept art

1.3.2 – Implementation

- Andrea:
    - Pre-flight page – airport information, routes, map and flight generator
    - Changed map used in all flight stages from Google Maps to leaflet
    - Styling of weather and flight status sections in Take-off, Climb, Cruise and Descent.
- Neil:
    - Connecting project to the simulator
    - Getting live data from the simulator
    - Displaying live data on project
    - Displaying aircraft location and direction on map
    - Calculations based on aircraft to determine current flight phase
    - Calculation of flight progress based on time
- Isaac
    - Connecting to API for airport autosuggest and markers component
    - Research of Third-party APIs used throughout the system
    - Calculating the fuel consumption and estimated time for a flight component
- Kyle
    - Creating and setting up the base application.
    - General style and layout of the web pages
    - User registration and Log-in system
    - Communicating with the Database and saving user data (previous flights and aircraft details)
    - Additional layers on map
    - Generation of route and connection to the flight plan database api.

- Parts of the Pre-flight page not mentioned prior (Online converter tools and Advances options)
- Change of autosuggest and markers from AirNav to JSON file
- Flight Profile, Flight Status, Airport Charts, Weather and Notepad components
- Runway, Metar, Airport Map and Airport Frequency components
- Mock flight data

1.3.3 – Final Report

- Joint
    - 2.5 – Flight Phases
- Andrea
    - 2.4 – Flight Planning
    - 3.1 – Requirements
- Neil
    - 2.2 – Flight Sim and Similar Solutions
    - 6.1 – Final Product
    - 6.2 – Product in Action
    - 7 – Conclusion and Final Work
- Isaac
    - 2.3 – Dividing into Flight Phases
    - 3.1 – Requirements
    - 3.2 – Analysis
    - 3.3 – Design
    - 5.4 – User Interface Design
    - 6.2 – Requirements Coverage
    - 6.3 – Testing
    - References
    - Appendix
- Kyle
    - 1 – Introduction
    - 2.1 – Initial Research
    - 2.2 (Section on Similar Solutions)
    - 2.6 – Questionnaire
    - 4 – Methodology
    - 5.1 – Framework Used
    - 5.2 – System Components and Architecture
    - 5.3 – Third Party APIs, Programs and Data
    - 6.1 – Final Product

1.3.4 – Additional tasks

- Kyle
    - Coordinator
    - Managing the Trello Doard
- Andrea
    - Meeting Logs
    - Figma Designs

2 – Research

2.1 – Initial Research

At the start of the project, none of us had any prior experience with flight simming or aviation in general. As a result, research was a crucial step in understanding the flight process. To start with, our mentor gave us a rundown of his process for flight simming [2], including some external websites that he uses and some basic terms in aviation [4-9].

The next step was to split the work as discussed in the prior section and conduct further research. Fortunately, there is a lot of information about aviation and flight simming online [4-6,10-11], particularly for Microsoft flight simulator. This includes online articles, forums and dedicated YouTube channels [12-16]. Our main findings are outlined in the subsections below.

After gathering information from various sources, we combined our findings by writing them down as part of a Trello board. This allowed for recording what will be needed in each individual phase, as specified prior. While this gave us a good idea of the general domain and what our project is required to accomplish, we still had a few specific questions. We decided that communicating with flight simmers through a questionnaire [see appendix 9.3] would be useful in order to better understand what they wish to gain from such an app.

2.2 – Flight Sims and Similar Solutions

MS Flight Simulator

When considering which flight simulator to work with we first looked at MS Flight Simulator. MSFS has a reputation for being one of the best flight simulators available and after some research it was evident that there were a variety of third-party applications working with MSFS.

Furthermore, MSFS has available 'SimConnectSDK', which is an official SDK that can be used to write add-on components that communicate with MSFS [17].

X-plane 11

When researching flight simulators 'X-Plane 11' kept popping up. X-Plane 11 is also regarded as being one of the best flight simulators available. However, when researching the simulator's developer tools, they seemed to be more focused on customization and creating planes [18].

In regards to other applications that offered similar features to ours we found many candidates but none were ideal.

For showing the airports, Airport visualiser [8] was our best inspiration. It is used primarily to gain information about airports worldwide. While this provides a lot of information, many of it is not useful for simmers. The main information needed was the airport code, elevation, location, runway configuration, frequency and navaids. In addition, for various aspects of an airport, the website redirects the user to another page, which is not ideal. We did find that the main user interface of a map with markers was easy to use. Sky Vector [19] also offered similar information, with the addition of airport zones and charts. However, not all information was accessible using the free version. Another option was Our Airport [9], however it offers too little data.

From our research, the most similar pre-existing solution is Flight Plan Database [4]. This can give most of the information needed by a simmer. Given two airports, it can create a route between them. It then shows the waypoints and jetway on a map, as well as a profile with the recommended altitude. It also shows the information about the airports, including the metar, runways, radio frequencies and navaids. It does still have limitations. The information is given all at once and requires a significant amount of scrolling. The flight profile is basic and not easy to follow. Not all necessary information is available on the website. Finally, there is no easy method to search for airports without knowing the code.

Another potential solution was LittleNavMap [20]. This offers an outstanding quantity of features for any situation. However, the interface is absurdly cluttered and outdated, and the setup process is tedious. This made the features available hard to find and use. It helped to emphasise the importance of the design and user experience considerations.

2.3 – Dividing into Flight Phases.

In order to keep the information presented to the user relevant during every part of the flight, the entire flight process has been split into multiple different smaller processes known as phases [3]. Each phase is defined by the different flight environments, information requirements and objectives. The separation of the flight process into phases helps greatly in managing the information to be displayed during the entire process, avoiding any unnecessary information to be displayed. The organisation of information by separating into phases improves the overall readability of the system since every piece of information on screen will be relevant to the particular phase on screen [21].

2.4 – Flight Planning

The idea behind having a pre-flight phase in the first place is so that pilots can have a carefully planned and detailed route to follow. The interface of this phase is, in a way, unique, compared to the other phases. This is because, while every other phase displays the aircraft and a map, this is the only phase designed in such a way that the map takes up the entirety of the screen and the aircraft does not yet appear (because the user has not entered the flight so the simulation has not started yet). We were inspired by Google Maps to take this approach – the first thing the user sees on Google Maps is a map showing their geographic location. This is done so the user can get a sense of what the function Google Maps provides, even though the user has not asked the map for information yet.

The main function of this stage is to plan a flight simulation. The user does so by inputting departing and destination airports which can be done by inputting the ICAO code or just by clicking on the airport icon of the airport itself and then clicking “Set Departure” or “Set Destination”. Once the two airports have been inputted, all the user has to do next is click the “Fly” button and the simulation starts. The application provides the user with a ready-planned route with all the necessary information readily available for them in the next flight stages.

The user can also opt to use our application simply to view the information of the airports. Once the user clicks on the icon of that corresponding airport, information regarding its region, charts, metar, runways and frequencies.

2.5 – Flight Phases

For the Taxi and Take-off phases, one of the primary sources used to learn more about these phases was online websites such as educational websites, online forums and YouTube videos [3, 22].

We learned that the Taxi phase is when the aircraft starts moving into take-off positions to depart from the airport. As such, one of the most important aspects of this phase is the selection of the runway. [23-24] Selecting the runway will determine which path the aircraft will take. We also learned how the speed of the aircraft is affected by the wind and that it should be taken into consideration when deciding on the distance the craft should take off when on the runway.

The Take-off phase takes place when the aircraft is going through the transition of moving on the ground to starting to fly in the air. As mentioned earlier, the speed of which the aircraft needs to Take-off is dependent on the current wind conditions. In this phase we also learned about the use of SID (Standard Instrument Departure) Charts [25]. An SID chart is a graphical illustration of the procedure to be followed from the take-off phase. This chart displays information which the crew of the aircraft have to comply with such as transition altitude when turning, navigation aid and bearings.

The primary sources of research for understanding the Climb, Cruise and Descent phases were online videos of simulations and forums [12-15,26]. This gave us a better understanding of what takes place in a regular flight, and what such an application would need to provide.

The first thing to understand was how altitude and flight levels worked, as they were crucial for this section of flight. The flight profile shows how the intended altitude of the plane should change with the distance covered. Planes fly within dedicated flight levels, with each level being at 1000 ft intervals. While these are typically assigned by ATC, simmers may choose to fly at any altitude if they wish [24-25, 27].

The waypoints were also an important part of flight that had to be considered. A waypoint is a fixed position with a set name. Each route is made of a series of waypoints from the departure to the destination. [24-25, 27]

The required frequencies for the radio were also different to those at the airport. The particular frequencies needed at any particular moment depends on the current flight information region (FIR) that the plane is in. Radio frequencies are very important to pilots as they can be used to communicate with ATC, check for weather conditions, and communicate with other flights.

Similar to the SIDs mentioned above, we also learned about the STARs (Standard Terminal Arrival) Charts [25]. These graphical illustrations provide information on how the flight crew should approach the landing of the arriving aircraft.

The Landing phase of a flight is in contrast to the take-off phase. In this phase the aircraft is moving from a low altitude to landing [28]. The aircraft will land on the designated runway by slowing down and descending onto it. In this phase we learned about ILSs (Instrument Landing System) which act as a guidance and in conjunction with radio signals from the tower and ground, they help the aircraft land safely.

2.6 – Questionnaire

From our research, we were able to get a good understanding of what information would be needed. We had also decided that the information displayed should be split into phases to avoid overwhelming the user. However, we were not confident about what information would be needed in each particular phase. We were also unsure about if both VFR and IFR flights need to be supported. Finally, we wanted to know what device would be used to run the app, as this would alter its layout. In order to investigate these, we decided to create a questionnaire [see appendix 9.3]. This was sent on multiple related forums. The main findings are outlined below:

Both VFR and IFR flight should be supported. While IFR is slightly more popular, both are rather popular and so both should be supported by the app.

Most simmers would either use a dedicated monitor or a tablet. This is crucial as it means that we do not need to support phone viewing for this prototype. Instead, we should focus on making a desktop-based layout. In addition, the fact that some simmers use a tablet means that a web app would be easier to access on various devices.

We were better able to sort the features into their ideal phases. The data collected helped to guide the design in later stages.

3 – Proposed solution

3.1 – Requirements

3.1.1 – Must-Have Features

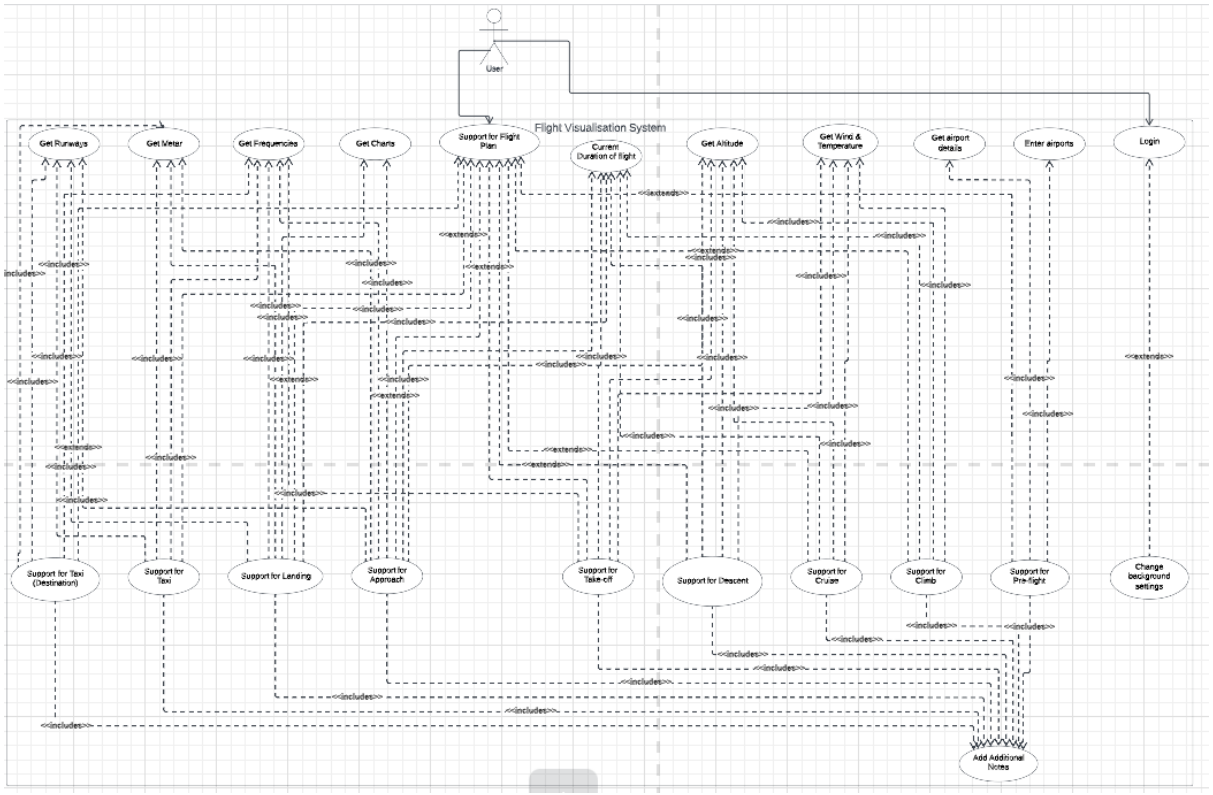
- Should work on desktop and tablets
- Runs on any browser
- Allows user to simulate any given flight
- Links to Flight Simulator
- Clearly indicates all phases of flight
- Clean UI – information should be displayed without any unnecessary clutter
- Information to be displayed is updated in real time
- All necessary Flight Charts such as SIDs, STARs
- Frequencies for Airport, Tower and Ground
- On screen visuals of current location of flight craft
- User Interface in general (i.e., charts, maps and graphs) should be easy to navigate through
- Map must not take too long to refresh and should refresh smoothly
- Where possible, icons should be used to describe information sections as opposed to text (especially for the weather section)
- Information displays which are to be used in multiple phases should always appear very similar

3.1.2 – Optional Features

- Flight Information Regions displayed on map
- Log-in system with customisable features
- Notepad for simmers to use

3.2 Analysis

The following Use Case Diagram shows the process of the system and how the user is going to interact with it when fully implemented:



The system's aim is to allow simmers to have a web-based solution readily available to them either on a specific monitor or personal tablet while simulating. One of the most important requirements for the system is to have a clean User Interface. When simulating a flight, simmers want to make sure that at their disposal there are the necessary details for the current phase of flight. Having a system which can automatically display the important pieces of details at the right key moments would save simmers vital time during the entire process.

Since certain pieces of information are always changing such as weather conditions, we deemed it necessary that all the information in display is displayed in real time, therefore the system will have to extract information from multiple online resources and make sure they are cohesive with one another to save the simmer time and also aid the simmer in making correct decisions during flight. Online resources are to be used to extract requirements such as the necessary charts for the system as well as important details like frequencies or the necessary fuel.

Since this is only to be used for a simulation, the decision to connect to Flight Simulator was taken in order to synchronise the mock up display of information on the web application with the simulation in Flight Simulator. The requirements of the system are geared towards achieving a viable working product that while still has minimum features, it can still be used to complete simulations of flight. However, if time permits the system also plans to include features which don't necessarily add to the information required in flights but are features that are nice to have and make the system much for customisable and robust such as the use of

different colour schemes or giving the simmer an option to write down temporary notes during the simulation.

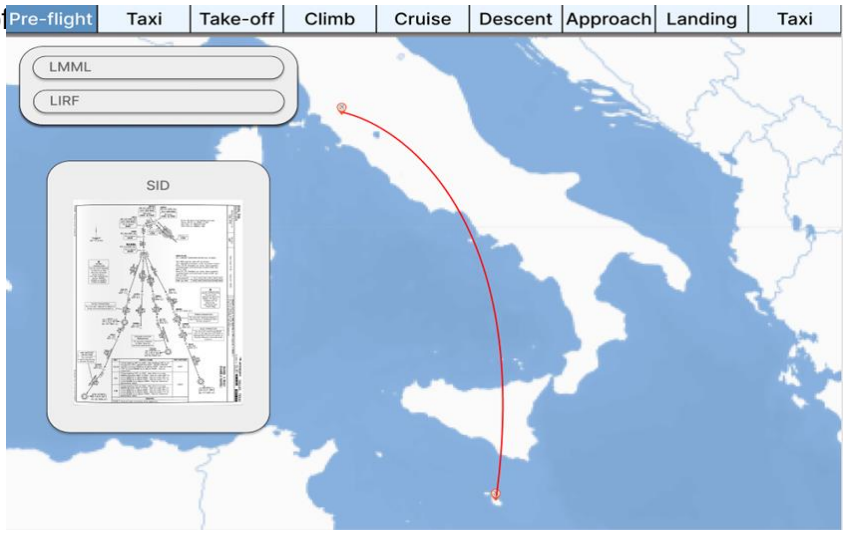
3.3 – Design

The design of the flight planning system is an important part of the system, users of the system will want as much information as required in the most clear and concise way, avoiding any unnecessary clutter in the system as well as having easy to read information on display at the right stages of the process.

The interview questionnaire, that was conducted during the planning stage of the process, was used to extract important information to determine what information flight simmers found to be valuable during the flight visualisation process. The responses from the questionnaire [see appendix 9.3] were taken into heavy consideration when deciding on the concept design of the system.

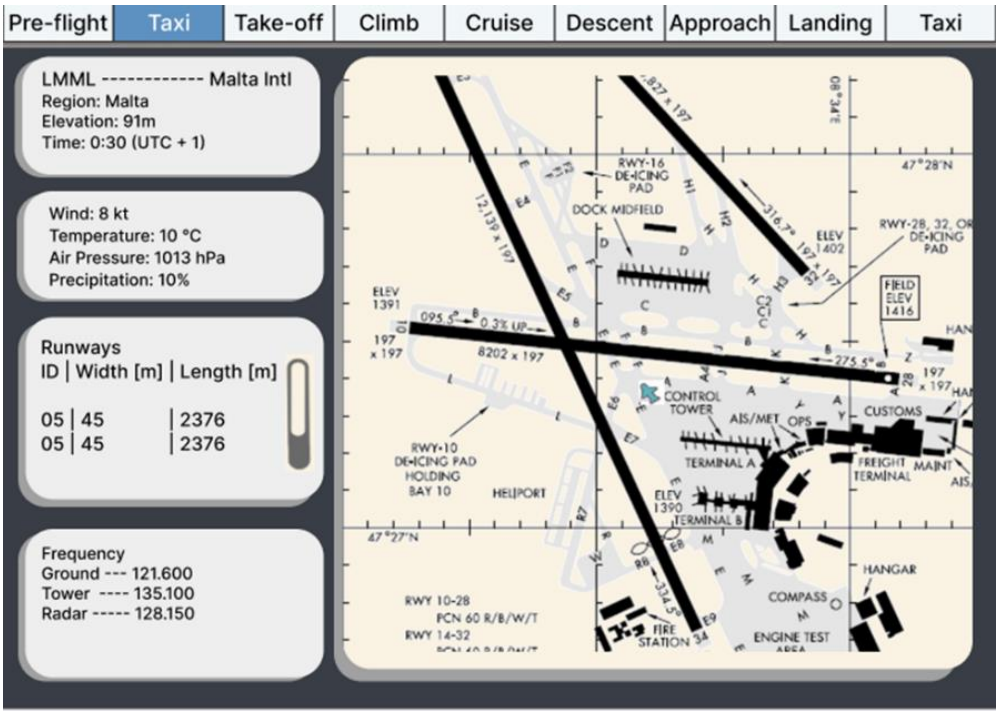
To represent each stage of the flight process, multiple tabs have been proposed to be added at the top of the screen to show which stage the user's flight is currently in and if needed be go back to previous stages with a simple click. The tabs of each stage would look similar to the image as shown below:

During the Pre-flight phase [see appendix 9.1], the starting point airports and the destination airport must be known and acquired, a preview of the flight path to be taken would also be beneficial in this section. Each airport has its own unique code to identify itself. In the below example, LMML which represents the Malta International Airport, is the departing airport, while the destination airport is LIRF, representing the Leonardo Da Vinci International Airport in Rome. The tab showing the chart will show various information relating to any airport the user decides to click, including its region, charts, metar, runways and frequencies. Having these details in the pre-flight phase is essential to allowing the user to run necessary checks needed before the phase of flight. Standard Instrument Departure Charts (SIDs) should also be present in this phase as they indicate the procedure on how a flight should proceed from the take-off of



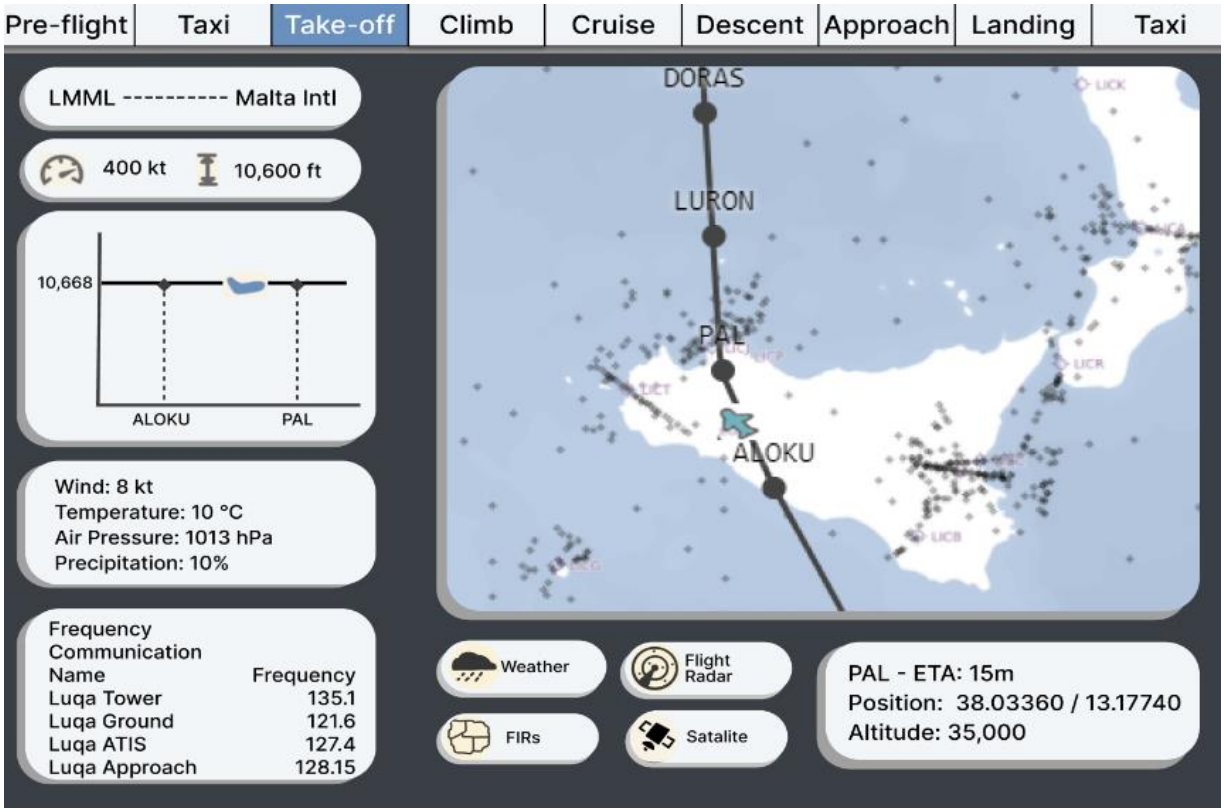
Proposed Design for Pre-flight phase showcasing the SID Chart

During both of the Taxi phases, a number of important details should displayed on screen with the main focus being the chart of the airport, as the chart of the airport. Since the aircraft is leaving or entering the airport, knowing the layout of the airport is vital information to have in this stage of the process as it allows for a smoother take-off or landing. Alongside the chart of the airport, other key information should displayed such as the profile of the airport (in this case being LMML) as well as the current elevation and also the wind and temperature. The wind has an effect on the aircraft as it can determine at what speed the aircraft should be moving at and how much runway it needs to use to take off. The available runways of the airport should also be displayed on screen as during this phase the user will select the runway that they will use to take-off. Details on the frequencies should also be available here as during these phase communication with Ground, Tower and Radar is important. All the information during this phase will be displayed as shown in the image below:



Proposed Design for Taxi phase

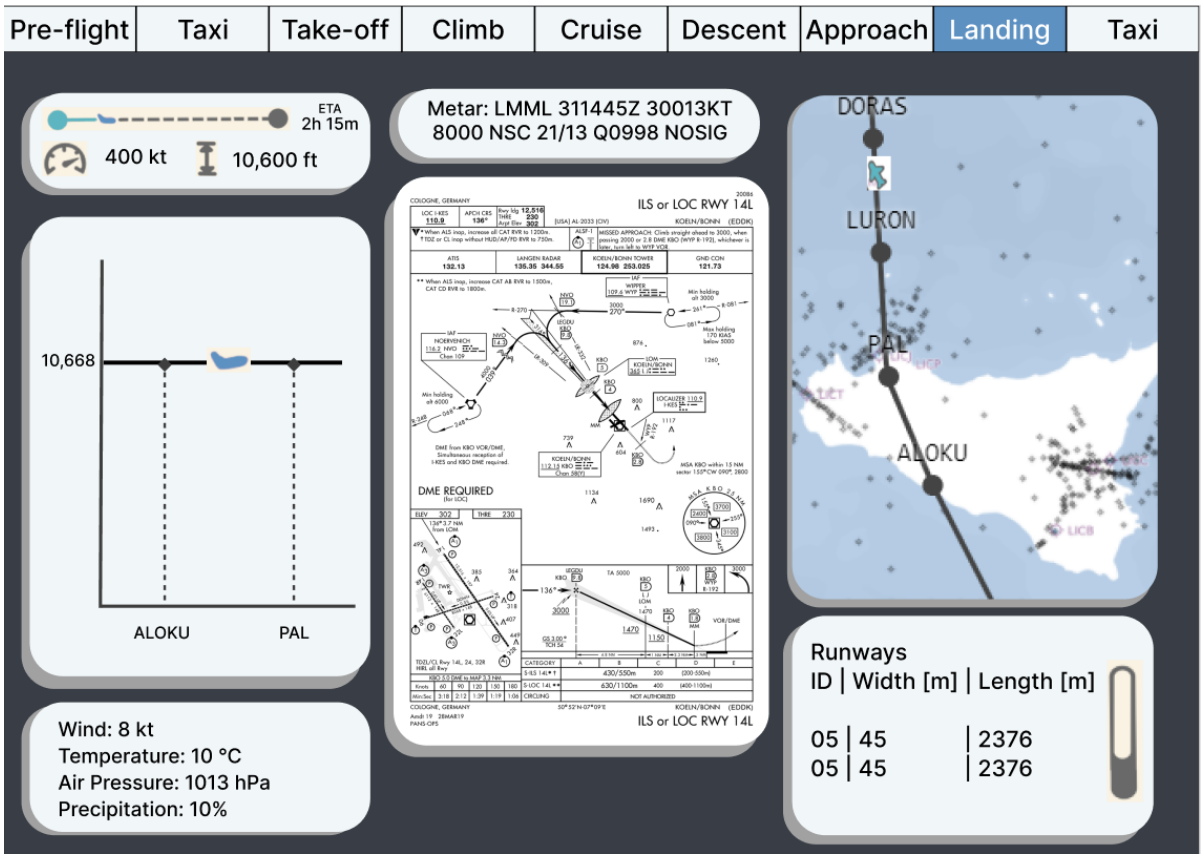
The next three phases are the Take-off, Climb and Cruise phases [see appendix 9.1], during these phases similar information will be displayed. During these phases, which take place when the aircraft is in the process of flight, it is important to display general details such as how much time remaining does the flight have until arrival and the route to be taken. Furthermore, real time trackers for details such as the speed, current altitude, weather, wind and air pressure are also important since decisions will be taken based on this information. Having access to the Flight Information Region (FIR) during this phase is also useful as it indicates which country is responsible for the operation control of the designated area.



Proposed Design for Take-off phase

During the Descent, Approach and Landing phases [see appendix 9.1], similar information to the previous phases should be displayed as the flight is still ongoing. However in addition now the crucial information to be displayed are the ILS chart as well as the STARS Chart. The ILS (Instrument Landing System) Chart is vital for this stage of the process as it provides an approach path for the descent of the aircraft on the final approach to a particular runway. On the other hand, the STAR (Standard Arrival Route) Chart is a standard route during the approach procedure for which an aircraft should proceed from the descent phase to the approach. During this phase the Metar should also displayed as the user would like to know the current conditions when approaching the destination in order to land safely.

The runways available and their current condition would also be of use in this phase as having the correct runway information available will give a better indication on how the aircraft should approach its landing.



Proposed Design for Landing Phase

4 – Methodology

4.1 – Agile Development

When deciding on the methodology, it was important to consider the particular constraints of this project. First, the presence of a strict deadline meant that we must be cautious not to commit to too many features, and have a managed scope. Second, as this project involves a field which none of us were familiar with, a significant portion of time should be dedicated to research, analysis and design. In addition, flexibility of changes during development is important.

As such, we decided to use an agile methodology. By having some requirements with lower importance, we were allowed to determine what we would be able to achieve during development. The flexibility of changes during development was useful as our understanding of the domain grew.

The specific agile methodology used was the scrum methodology. This consists of short sprints of work after which a result is outputted. In order to keep track of this, a Trello board was utilised. This allowed us to monitor which tasks were to be worked on and which were completed. In addition, regular meetings were held. Typically, we would have a short meeting between team members at the start of the sprint and longer meeting with our supervisor at the end of the sprint. The sprints were typically one week long during the research and design phases, and two weeks long during development. This system allowed for all team members to work on their own tasks individually, before combining the results together.

While we had set milestones and deadlines for the different phases of development at the start, these proved optimistic. Both the requirement elicitation and the design took longer then expected, which was likely due to our limited knowledge of the domain. As such, the development had to be done at a faster pace.

4.2 – Tools Utilised

As this project involved working as a group, coordination and communication was very important. In order to achieve this, various tools were utilised.

As mentioned prior, Trello was used as a task board to keep track of what we are trying to achieve. By adding deadlines and team members to a particular task, it was easy to identify what any person should be working on and if they are progressing as they should be. In addition to task management, we found that Trello worked well for keeping track of the data that a simmer would require. This was because the list structure could be used to visually represent the phases of flight.

Google docs was heavily utilised. First of all, the final report was written in this program so that all team members can simultaneously work on it and identify the progress being made. In addition, another document was used to store the links to any other shared documents or resources. We found that this system was a low maintenance way to keep all necessary information easily accessible, without having to keep track of multiple lists.

For general communication, a messenger group chat was used. This allowed for notifying team members of tasks and organising the next meeting. As some team members were from different courses with different schedules, having an asynchronous and online method of communication was vital. Both messenger and zoom were used for online meetings, although zoom was preferred for its better shared screens feature and ability to record meetings.

For eliciting requirements, google forms was used to construct the questionnaire. We found this program well suited for our needs, and easy to utilise. In particular, the data visualisation after the responses were gathered was very useful in the later stages.

During the design phase, Figma was used to design the layout of the application. The main benefit of this was the ease of collaboration between members. Initially, a general-purpose drawing application was used instead, however this proved inefficient for multiple people to use and ill-suited for our purpose.

Finally, Git and GitHub were utilised to share the project code between members. These technologies are an industry standard, and vital for collaboration. While fixing conflicts when merging was occasionally an issue, it is well documented with a large community, so support can always be found.

5 – Implementation

5.1 – Framework Used

A framework should be used in the development of the application. There are several benefits of this. Using a framework can reduce the workload by providing certain pre-existing features, such as a login system. They can also provide a more organised structure and architecture that follows a well-defined pattern.

When determining which framework to use, there are several factors to consider. A good framework should provide any standard features required, while having the least amount of bloat possible. The learning curve that needs to be overcome and the experience of our team must be considered, as the development time is limited. In addition, having good documentation and community support is one of the most important aspects. Troubleshooting is significantly harder with limited support.

Four frameworks were considered as potential options: Angular, Laravel, CakePHP and Asp.Net Core. Angular was a viable option. It has extensive support for auto-updating applications with repeating components. However, the learning curve was considered too steep given the experience of our team. Two PHP based frameworks were considered, namely Laravel and CakePHP. Laravel had limited functionality for our purposes, and CakePHP did not have sufficient documentation, so neither were chosen.

In the end, Asp.Net Core was chosen as the framework to be used. From our research, we found that it had the necessary features. The documentation and online support is excellent due to its maturity and age. Performance and bloat were considered well balanced for our requirements. By building on the base application that is generated by default, a significant amount of time was saved.

5.2 – System Components and Architecture

The Asp.Net Core framework uses the MVC (Model, View, Controller) pattern. This pattern is common in web applications. As the name suggests, there are three important components in the pattern, each managing different areas. The separation of concerns allows for less coupling within the application, which makes maintainability easier. It is also a modular system that helps to avoid repetition of code.

Models are classes used to store and format any data and application state. There can be different types of models depending on what the required purpose is. In addition, a repository can be used to get the data from a database or API, which would populate the model. Views manage the user interface. In the case of a web application, this means the HTML page and style that the user will interact with. The controller class is used to store any business logic. These classes link the views and models together by handling routing and passing the data to the pages.

For our application, interactions with APIs are very prominent, as we are gathering data from many different sources. These are handled in different repositories, and will be specified in detail in the upcoming sections. By having repositories implementing an interface, it is possible to change the API without having to make any changes to the views.

Each separate phase of flight has its own view. The Layout page is common to all pages and provides the header bar. In addition ViewComponents were heavily used throughout the project. This allowed for having certain common components that can be used throughout the app multiple times. Each ViewComponent has its own style and javascript file, which makes it easier to insert them into different pages.

A key property of the web application is that it updates live alongside the flight simulator. To achieve this, JavaScript was used by the different components. Ajax calls are made at regular intervals to the controllers to get the required data, and update the page accordingly. While there are multiple controllers linked to certain aspects of data, the main controller is the PhaseController. This controller is used to set up and read data related to the flight and route.

While it is not a required component of the application, there is a login system. This was achieved by scaffolding Microsoft's Identity component. This allowed for a complex, safe and robust authentication system. This was linked to a PostgreSQL database using the entity framework to store details of the user, and any persistent data. In order to allow the application to work without requiring logging in, the route is stored as a session instead of in the database.

5.3 – Third Party APIs, Programs and Data

Many different third-party applications were used throughout the project. This section aims to catalogue and briefly explain the purpose of all external data sources used.

The most prominent API used is that from Flight Plan Database [4]. This is used to generate the flight route and profile. In addition, it is used to get the details about the airports for the Pre-flight and Taxi phases. From our experiments, the API works with most required airports. There were some limitations with the data gathered from this source. The primary one being that the flight profile is rather basic as a straight line is used for the climb and descent. Another issue is that the required volume of fuel is not provided. Another API from despouy.ca [29] was used for this purpose.

To connect and retrieve data from the simulator we made of the 'Microsoft Flight Simulator Software Development Kit'. This is referenced as a DLL file within the application. The main data points gathered from the flight simulator are the altitude, position and speed of the aircraft. Not only does this SDK allow our project to connect with the simulator, but in the documentation, a program with the name of 'SimvarWatcher' is present. This SimvarWatcher and amended as necessary to allow the project to retrieve live data from the simulator [30].

For displaying the map, Leaflet [31] was used. This is a free to use mapping system that we used extensively throughout the application. The map data comes from Openstreetmap [32] and Openweather [33]. An additional plugin was used to rotate the plane marker. Initially, Google maps was used to show the map, however this was changed as the free version was not deemed sufficient.

For the FIRs, the data and code was adapted from the following GitHub Repository [34]. A limitation of this source is that most FIRs are rather simplified outside of Europe and North America.

For the graph of the flight profile, Chart.js was used to create the graph. While some of the formatting options were rather cumbersome, we found that it was capable of achieving what we require.

In the Pre-flight page, the airport markers and name autosuggest are based on a JSON file that is openly available on github [35]. Initially, an API from AirLabs [36] was used, however we found that the request limit was too low for our needs.

OpenWeather [33] is used to get the weather at any given position. While this worked well for demonstrating the concept for the sake of a prototype, this API does not take the altitude into account. As such, a different data source may be a better option.

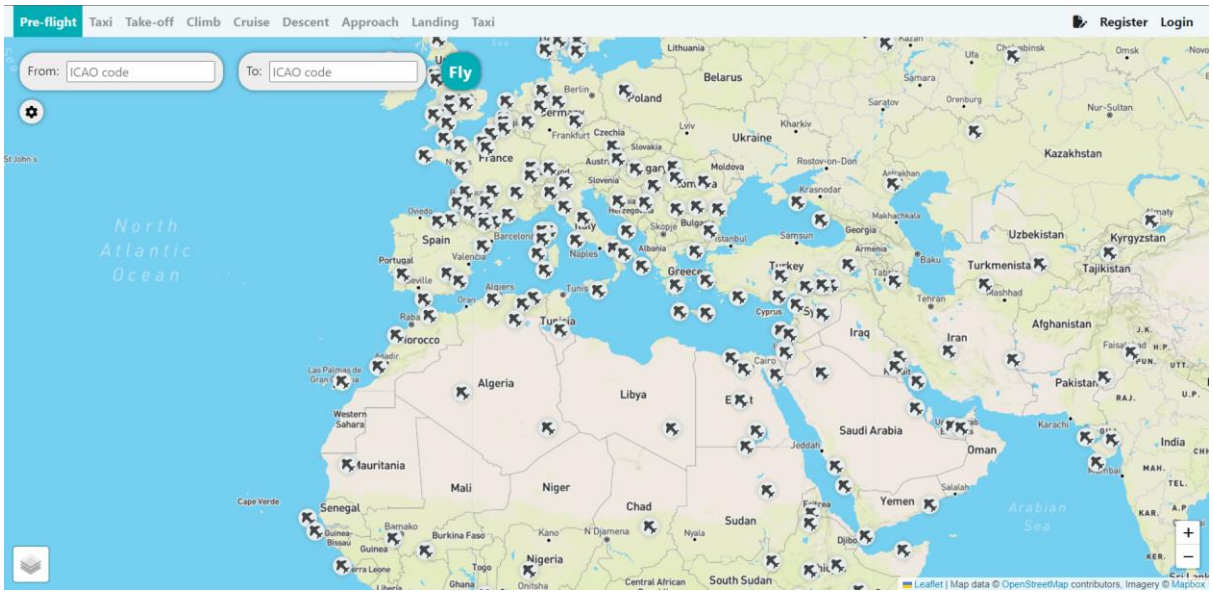
For the charts of the airport, the website Universal Airlines was used. While this was the best free option available that we found, it does have various limitations. The charts provided are rather cluttered and require significant scrolling, which is opposed to the purpose of our application. In addition, it is rather slow to access which degrades the user experience. Initially, we were planning to use ChartFox [37] as an API to access the airport charts. This free

resource was more user friendly and is faster to access. Unfortunately, the API tokens are only given on request and we were unable to contact ChartFox despite multiple attempts.

5.4 – User interface design

The User Interface was created with the idea of being usable but also having the necessary details available to the user when necessary. During the research phase of the process, we noticed that many systems with similar features were lacking in cleanliness [21, 24-25].In the system we tried to use as many user friendly features based on best practices and input from simmers [see appendix 9.3].

The user is greeted with a world map with multiple small aeroplane icons on screen, signalling the users an airport. With just a click the user can obtain a wealth of necessary information regarding the selected airport. This feature allows users to have an approximate idea of their desired flight and also gives the user time to take note of any details they might need before taking flight.



Pre-flight Start-up page

The map is overlaid by 2 features, the settings feature as well as the more prominent panel, the flight panel asks for the input of which airports would be the start and which airport will be the destination, this feature is used to initialise the process of flight. The settings feature includes multiple advanced features such as speed and rate of ascension. The aforementioned features are not details which the knows or necessarily is aware of therefore it was decided to contain these advanced features in a settings button.

The top panel, also the most crucial panel, features the flight phases – each phase is marked in chronological order from left to right, indicating the start and end of the flight process. Initially this section is mostly greyed out due to the flight process still being in the pre-flight phase, which is highlighted on the panel. Once the flight process begins however [see appendix 9.2], these panels are activated as indicated by the colours used. This decision was taken to make sure that the user is led into starting the flight process, if the system allowed users to access other phases, the user could end up getting lost and not realise how to start the process. To give the user an idea on what phase of flight they are in at the moment, a small indicator next to the corresponding button is displayed in the form of a plane. This marker changes as the

flight process moves on to the following phases. On the world map, a mock of the route is displayed showing all the waypoints the system will pass through during the flight.

In the flight process pages [see appendix 9.2], the page's screen is split into multiple different components on the screen, each indicating a useful piece of information to be used during that particular phase. The layout of the page is extremely similar to one another as it will allow the user to much more quickly identify where the information they need is during the flight process. If the layout of these pages were significantly different from one another then the user would have to adjust and study the page to understand were the information is. Using a more similar layout saves time for both the users and also in developing the system. Each component in these pages are also of a particular size, the bigger they are in size the more vital they are to the particular phase. This decision was taken as generally the most important pieces of information of the phase is what users want the most, therefore making this information the largest would also save time in scanning the page to find it as the information is instantly recognisable thanks to its presence on the screen.

During the flight phases some of the general information has been given a visual representation instead of simple text to make it more readable. For starters, the duration of flight is indicated by a progress bar which fills up the closer the flight is to destination. Moreover, information for weather and wind for example has been iconised to take less space in the system. Altitude of the flight is also graphically represented on screen to show the user the altitude levels during the process.

6 – Evaluation

6.1 – Final Product

Our goal for this project was to create a proof of concept for an application that can be used alongside a flight simulator for easier access of information. In this regard, we believe that we were successful. The final product can be used with most flight routes and should be capable of displaying the most important information at the point that it is needed by the simmers.

An important concept in software development is that each step is well linked to the next. This concept was upheld in our project. The requirements for the application were based on the research and questionnaire conducted. The concept design focused on fulfilling those requirements in the best possible way. Finally, the implementation closely follows the design. This ensures that the final product is something that simmers would find useful.

There were some changes made from design to implementation. Some of these were due to further research and understanding, while others were due to time constraints. As an example of the former, we found that the FIR frequencies were not of use to most simmers, so we dedicated the screen space to a larger flight profile and instead allowed for users to display the FIRs on the map instead.

Regarding features that we were unable to implement due to time constraints, the ability to reroute the plane mid-flight was one such example. As we followed the agile methodology and not all the requirements were essential, this is allowed. The alternate route was deemed to be non-essential and so it was pushed back. We were still able to add various non-essential features, such as saving previous routes and aircrafts to a database.

Another limitation the team faced was a lack of knowledge on how to operate aircrafts in the simulator. Due to experience and time constraints all testing was done using an aircraft piloted by an AI. It is unclear if any bugs are present if the aircraft is being piloted by the user, particularly in the logic behind how the project determines in which phase the aircraft is. Furthermore, the user must enter a flight path in the simulator for the project to work as intended.

There are two prominent flaws with our system. The first is a lack of error handling. Invalid input by the user can cause issues with the system. If no charts are found, a 404 is shown instead. The second is regarding maintainability and coupling of the code. For some repositories, a JSON string is returned instead of a set object. This JSON is then parsed in the JavaScript file and causes unnecessary coupling. The effect of this was felt when we changed the airport list system from Flight Labs API to a JSON file. As we were unfamiliar with ASP.Net Core and the MVC pattern, we were not aware of this best practice.

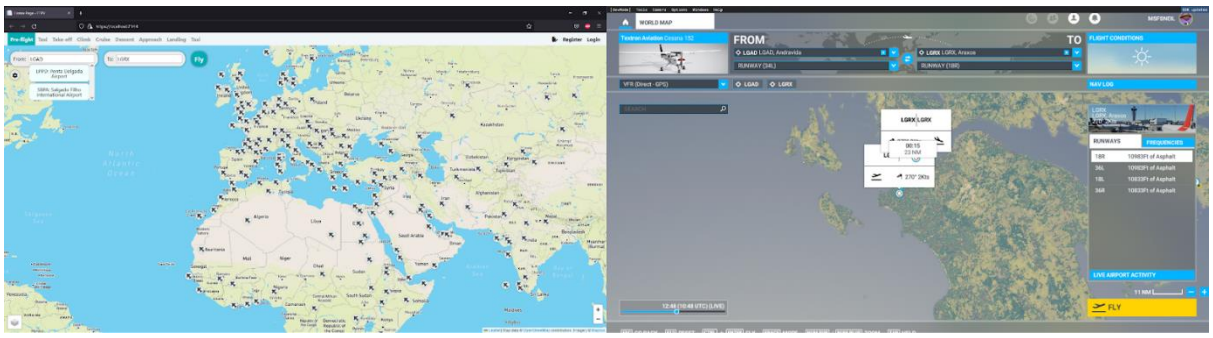
For the final evaluation, we wished to meet up with various simmers who could use the prototype alongside a real flight. This would give us concrete feedback and a better understanding on if all requirements were fulfilled. Unfortunately, we were unable to do this due to time constraints.

6.2 – Program in Action

This section will explain how our project interacts with ‘Microsoft Flight Simulator 2020’

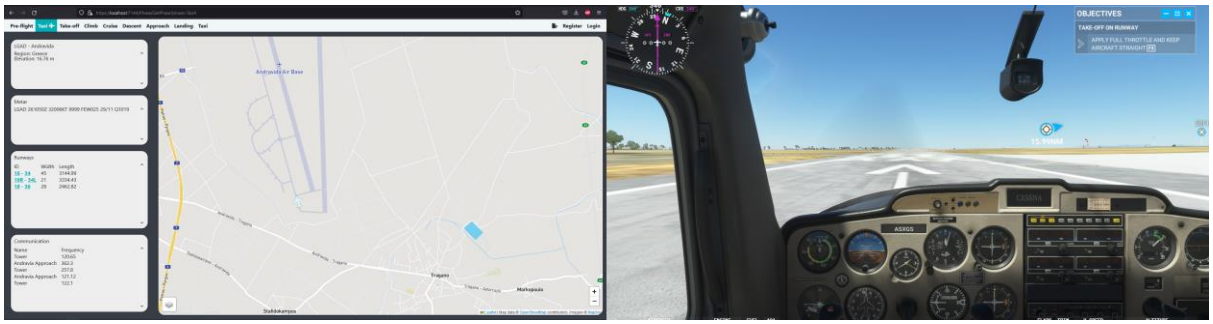
Start

The user enters their desired departure and arrival airport in both the simulator and the project. It is also important for the user to click on the settings icon from the project and tick “Display live data from a simulator”. This feature is important as it is what instructs the project to acquire live data from the simulator.



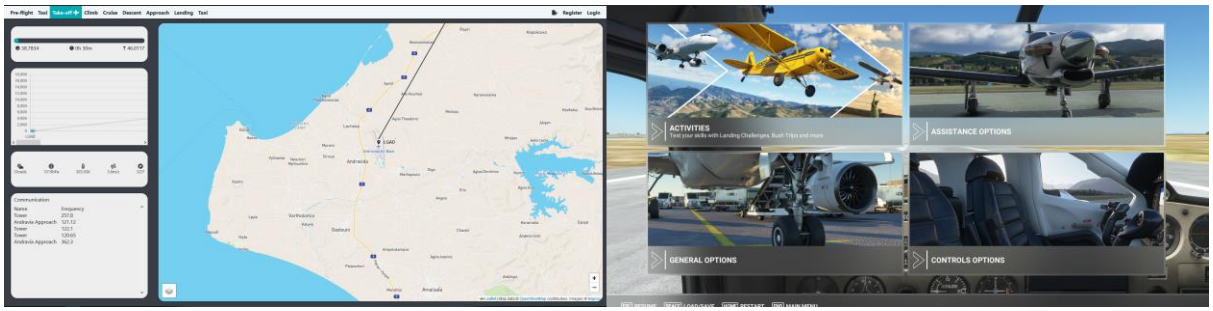
Map

The project obtains live data from the simulator regarding the aircraft’s location and direction, and it is accurately displayed on the map.



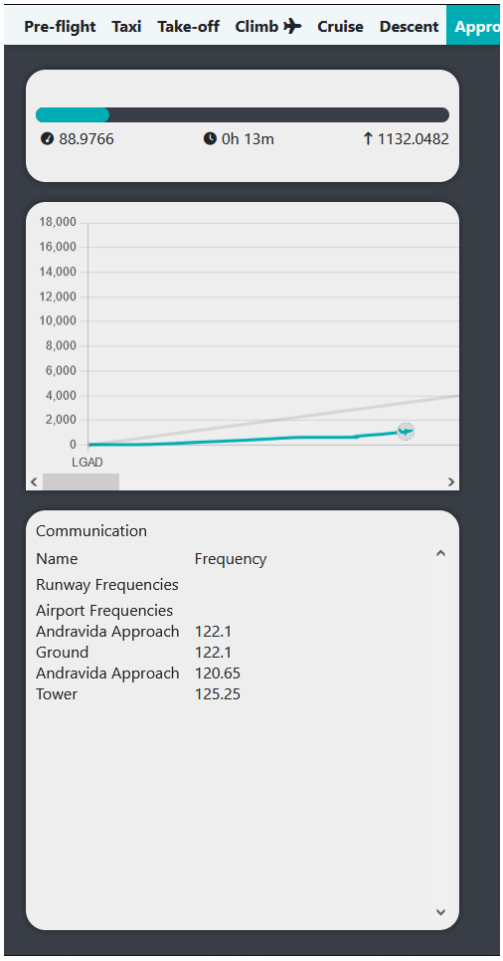
Switching of Phases

Through the live data that the project is obtaining from the simulator and a series of mathematical calculations, the project determines the flight phase that the user is currently on. The user can opt to manually switch between phases depending on their preference, or leave everything to the project, which will show the user the flight path that they are currently on.



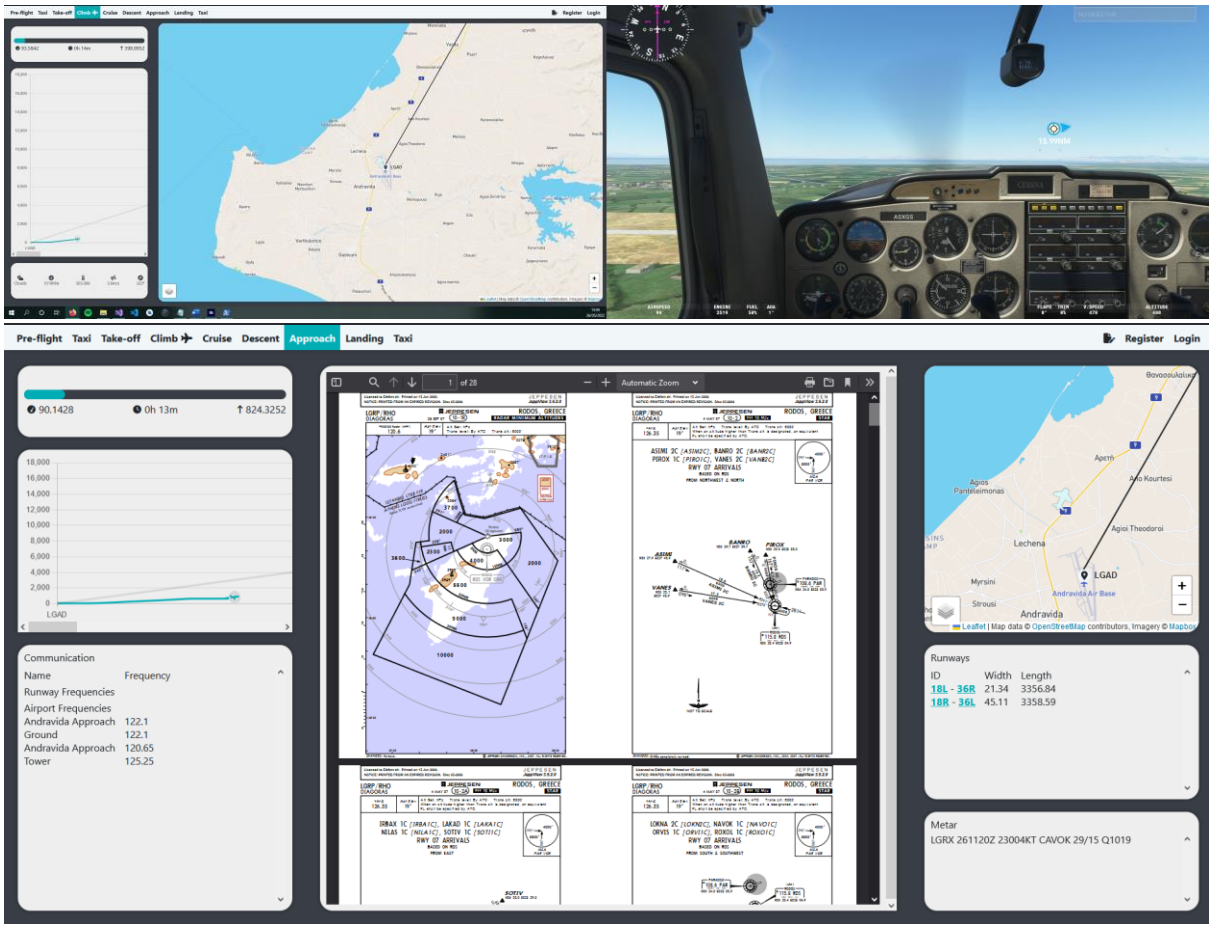
Progress Bar

During certain phases a progress bar is visible to the user. This progress bar is calculated based on the remaining time and time spent flying. The progress bar is meant to show the user in a graphical way the percentage completed of the flight.



Displaying Data

Data which was deemed key for the user (speed, altitude, Metar, etc.), is displayed to the user in the phases in which it is deemed to be needed. The data is updated depending on the aircraft status in the simulator and the departure and arrival airports.



6.3 – Requirements Coverage

6.3.1 – Coverage of Must Haves Features

Should work on desktop and tablets: Web application is fully functional on desktop and tablet hardware through the use of browsers

Runs on any browser: System runs on all popular browsers available such as Chromium-based browsers like Google Chrome and Microsoft Edge as well as other popular browsers on different engines such as Firefox and Safari.

Allows users to simulate any given flight: Users can simulate any flight from any airport of their choosing given that the option is listed and available from the website. No occurrences of any missing airports were detected.

Links to Flight Simulator: User can link to Flight simulator through the settings function by ticking the “Connect to Flight Simulator” feature.

Clearly indicates all phases of flight: All phases of flight are clearly marked on screen and the system has a real time marker which indicates which phase of flight the process is currently in.

Clean UI – information should be displayed without any unnecessary clutter: Information on screen is neatly presented and separated into their corresponding phases. Information is well spaced out to avoid unnecessary clutter.

Let users make changes to route in case of emergency: User can decide to end his currently selected route in the Pre-flight phase and restart.

Information to be displayed is updated in real time: The process of flight is updated in real time to indicate which phase the system is currently in. Individual components such as time, altitude and weather are updated in real time.

All necessary Flight Charts such as SIDs, STARs, chart of Airport: Flight Charts are available in the system through a PDF display however in this format all the charts are displayed and not separated, furthermore due to limitations of the third party APIs used, some of the airports do not have any charts available

Frequencies for Airport, Tower and Ground: Communication Frequencies are displayed in the necessary phases of flight such as the Taxi phase.

On screen visuals of current location of flight craft: A mockup of the flight is updated in real time on the world map available. A pointer in the form of an aircraft is displayed on screen and this pointer is constantly moving towards the destination.

User Interface should be easy to navigate through: Interface requires very little user input to make use out of it. To initiate a flight, the user needs to simply input the desired flight and confirm. To access information of other phases of flight all is required is one simple click of a button that corresponds to the desired phase.

6.3.2 – Coverage of Optional Features

Flight Information Regions displayed on map: A layer of the FIRs can be displayed on the world map available

Log-in system with customisable features: If the user is logged in, the details of the aircraft and the list of previous routes are saved. This avoids having to re-enter the same details over and over. No further customization was added due to time constraints.

Notepad for simmers to use: Simmers can take down, edit or delete notes simply by clicking on the note button on the top right.

6.4 – Testing

Due to time constraints, we were unable to contact a simmer to test the flight visualisation system, a tester would have been helpful to identify the use of the current system.

In order to test the usage of the current system, it is important to look into the market and see what systems are available that have a similar usage to our system and determine whether the system is comparable. In section 2.2, we identified a number of systems which we took inspiration from when researching and designing the system but no that the system is complete, how does it compare to the applications?

One of the most popular applications is Flight Plan Database [4], this application's API was used extensively during the creation of our app as the database has some of the most up to date data that is easily available to the public. While Flight Plan database is an extremely useful app which has data available that is not present in our system such as some extra details on the runway, the website takes a long time to navigate through despite. While the Flight Plan Database similar to our system has information presented neatly, the information takes a long time to get to as the system is not organised into different phases and is simply included in one page which you have to scroll through to find the information required.

The other map that was analysed extensively during the research and development of our application was the LittleNavMap application [20]. LittleNavMap boasts an incredible wealth of information to the user that exceeds even the amount of information that is available in the Flight Plan Database so in comparison to our system, LittleNavMap provides the greater amount of details. However, LittleNavMap while does provide a wealth of information, it is extremely difficult to navigate through and also learn where everything is. Since the system has so much information, the application sacrificed readability in order to achieve possibly one of the most content rich systems available to the public. When compared to our system, it is clear that our system in comparison is much easier to learn and navigate through. While LittleNavMap provides almost anything a user could ever need, our system has makes sure thanks to extensive research and the interview questionnaire that the most crucial pieces of information required by the simmers is still available and accessible to the users.

One of the primary goals of the system in the requirements was to create an easy and user friendly system that can allow the simmers to make faster decision during the flight process since we felt that there was a clear lack of a system with a easy to read user interface. While other systems such as the aforementioned LittleNavMap contain more information it is important to note that these systems have access to the latest information and have received financial support to gain access to this information. During our development of the system, in some cases such as the use of charts, we had to make use of the most easily accessible options as a lot of flight information can only be accessed by purchasing the information of the developers.

7 – Conclusion and Future Work

From our evaluation, we consider the product to successfully satisfy the given task. According to our research, this application should provide all the basic information that is required by a simmer in as efficient and user-friendly manner as possible. While some minor features were not achieved, overall, most of what we set out to achieve can be seen in the prototype.

For future work, additional features could be added to cater for more niche situations and needs. This however should not be at the expense of having a cluttered interface, as this would invalidate the benefits of the application. In addition, better APIs and data sources could be found or perhaps purchased, particularly for the airport charts. A future improvement for flight charts would be the ability to call a particular chart as well as downloading a specific chart.

In its present state, the project only connects with the simulator if they are both on the same devices. This limits the user as it cannot be used on a tablet or laptop. An additionally future work could be to allow the possibility for users to connect to a simulator on a separate device from where the project is being run.

8 – References

[1]"FlightSim Community Survey Results 2021 - FSElite", *FSElite*, 2021. [Online]. Available: https://fselite.net/content/flightsim-community-survey-results-2021/.

[2]E. Cachia, "GAPT Meeting 23rd February", 2022. [Online]. Available: https://universityofmalta.zoom.us/rec/play/xtnH7tKGMVKyjIyzHhFgbeAAqXh3-cqZSGU16da-p3YaUlAfgngaLQFog1Au2ZfL6nM_byPYyf3gb1pw.YXkj8kO-IQTb0HlV?continueMode=true.

[3]"Phases of Flight". [Online]. Available: https://www.infohas.ma/en/phases-of-flight/

[4]"Flight Planner", *Flightplandatabase.com*. [Online]. Available: https://flightplandatabase.com/planner.

[5]"Online Flight Planner", *Onlineflightplanner.org*. [Online]. Available: http://onlineflightplanner.org/.

[6]"SimBrief.com - Virtual Flight Planning Solutions", *Simbrief.com*. [Online]. Available: https://www.simbrief.com/home/.

[7]"Flight Calculator", *Travelmath.com*. [Online]. Available: https://www.travelmath.com/flights/.

[8]"Airport Visualizer - Worldwide Airports Information on 30+ Maps", *Maththinking.com*. [Online]. Available: https://maththinking.com/airportvisualizer/.

[9]"Map of airports in the World @ OurAirports", *Ourairports.com*. [Online]. Available: https://ourairports.com/world.html.

[10]"RouteFinder - Flight planning tools", *Rfinder.asalink.net*. [Online]. Available: http://rfinder.asalink.net/. [Accessed: 24- May- 2022].

[11]"Live Flight Tracker - Real-Time Flight Tracker Map | Flightradar24", *Flightradar24*. [Online]. Available: https://www.flightradar24.com/.

[12]"FULL FLIGHT with ATC! Microsoft Flight Simulator - Citation CJ4 (VATSIM)", *Youtube.com*, 2021. [Online]. Available: https://www.youtube.com/watch?v=7Q-M4XVB_is.

[13]"✈ How Do You Create Flight Plans in Microsoft Flight Simulator 2020? ✈", *Youtube.com*, 2020. [Online]. Available: https://www.youtube.com/watch?v=yiaDG5HWBHU.

[14]"Microsoft Flight Simulator | XBOX | Flight Planning Made Easy || Getting Started Guide", *Youtube.com*, 2021. [Online]. Available: https://www.youtube.com/watch?v=JejFa3y-vvk.

[15]J. Parkin, "Microsoft Flight Simulator guide: How to create a flight plan", *Polygon*, 2020. [Online]. Available: https://www.polygon.com/microsoft-flight-simulator-guide/21402724/flight-plan-departure-arrival-waypoint-map-add-vfr-ifr-vor-compass.

[16]"Flight Simulator 2020 Flight LESSONS | HOW TO GET STARTED | Pilot Teaches How to FLY - Tutorial #1", *Youtube.com*, 2020. [Online]. Available: https://www.youtube.com/watch?v=GvBG7pVhRBE.

[17]"SimConnect SDK", *Docs.flightsimulator.com*. [Online]. Available: https://docs.flightsimulator.com/html/Programming_Tools/SimConnect/SimConnect_SDK.htm.

[18]"Development Tools | X-Plane", *X-Plane*. [Online]. Available: https://www.x-plane.com/desktop/extend-it/development-tools/.

[19]"SkyVector: Flight Planning / Aeronautical Charts", *Skyvector.com*. [Online]. Available: https://skyvector.com/.

[20]"Little Navmap", *Littlenavmap.org*. [Online]. Available: https://www.littlenavmap.org/. [Accessed: 24- May- 2022].

[21]E. Cachia, "GAPT Meeting 2nd March", 2022. [Online]. Available: https://universityofmalta.zoom.us/rec/play/cjInR0cTezarljBSKPQTkmfuXs8JjBoAch8W0jKVgiXQWy3AwdmZAgjHIeq2ce6V530P6xjJkEIlno69.NNIe7BS_RE54Uip3?continueMode=true.

[22]"Select the runway for take-off and landing - IVAO - International Virtual Aviation Organisation", *Mediawiki.ivao.aero*. [Online]. Available: https://mediawiki.ivao.aero/index.php?title=Select_the_runway_for_take-off_and_landing#:~:text=The%20term%20%E2%80%9Crunway-in-,choice%20of%20the%20active%20runways.

[23]E. Cachia, "GAPT Meeting 10th March", 2022. [Online]. Available: https://universityofmalta.zoom.us/rec/play/xh17hKrTTvpGUPhXZA_g1B_L1ESodDkllbmYw8KmqgZe90g-pZaRwvslf6KYFWF85o05bQk576SB3ZW3.b8WPGq-VtFcXU6rx?continueMode=true.

[24]E. Cachia, "GAPT Meeting 18th March", 2022. [Online]. Available: https://universityofmalta.zoom.us/rec/play/vcIXCKMwOOi13QR1-BcW9u_nFSTCfjLGiwgyAzK4shzJfoTKJhhli3-UmGTKBJQ2Aq5GbRaaT2F1svN7.NFhZx9kZyy6bn8gj?continueMode=true.

[25]E. Cachia, "GAPT Meeting 25th March", 2022. [Online]. Available: https://universityofmalta.zoom.us/rec/play/X9wvtygp9-5wmRAyXJlgb0T765qbXoB6SG4PoxlSI7MJgDjO9nx1ssXrzZNVagqZ5eH7Rb9DwOQ0NOs.AbP6g7aMqbo3kpH1?continueMode=true.

[26]C. Page, "What do pilots do during the cruise? - The Points Guy", *The Points Guy*, 2019. [Online]. Available: https://thepointsguy.com/news/what-do-pilots-do-during-cruise/.

[27]E. Cachia, "GAPT Meeting 30th March", 2022. [Online]. Available: https://universityofmalta.zoom.us/rec/play/ZIeq84BsyZMk_R-2pzMnf3prZJ218Q2L30-3EixxnJ3l-TQOp6vbJzMRDB18D_1xFV1jeLzM-xCVKSI.O5tLzQiyZQJRJuSJ?continueMode=true.

[28]"Landing distance - IVAO - International Virtual Aviation Organisation", *Mediawiki.ivao.aero*. [Online]. Available: https://mediawiki.ivao.aero/index.php?title=Landing_distance.

[29]"Pedro Despouy", *Pedro Despouy*. [Online]. Available: https://despouy.ca/.

[30]"Introduction", *Docs.flightsimulator.com*. [Online]. Available: https://docs.flightsimulator.com/html/Introduction/Introduction.htm.

[31]"Leaflet — an open-source JavaScript library for interactive maps", *Leafletjs.com*. [Online]. Available: https://leafletjs.com/.

[32]"OpenStreetMap", *OpenStreetMap*. [Online]. Available: https://www.openstreetmap.org/#map=11/35.9448/14.3810.

[33]"Current weather and forecast - OpenWeatherMap", *Openweathermap.org*. [Online]. Available: https://openweathermap.org/.

[34]"GitHub - jaluebbe/FlightMapEuropeSimple: Aircraft positions received by OpenSky Network contributors as well as airports and airspaces are shown on a map.", *GitHub*, 2019. [Online]. Available: https://github.com/jaluebbe/FlightMapEuropeSimple. [Accessed: 24- May- 2022].

[35]"JSON data for airports and their locations", *Gist*. [Online]. Available: https://gist.github.com/tdreyno/4278655. [Accessed: 24- May- 2022].

[36]"Air Quality Management - AirLabs", *AirLabs*. [Online]. Available: https://www.airlabs.com/.

[37]"ChartFox - Free Flight Simulation Charts", *Chartfox.org*. [Online]. Available: https://chartfox.org/.

[38]"Airport Layout", *Youtube.com*, 2020. [Online]. Available: https://www.youtube.com/watch?v=ebHlrtw0oLM.

[39]"Airports - 01 - Airport Markings", *Youtube.com*, 2016. [Online]. Available: https://www.youtube.com/watch?v=MrrL92AH23A.

[40]E. Cachia, "GAPT Meeting 7th April", 2022. [Online]. Available: https://universityofmalta.zoom.us/rec/play/SpnBpVaFpUaQeS0lhBBVFdMNSHx87PHmmHJAJGSZlb_UntpqbI4nsye92GjyjRkFD4IMGJKoGUtENk5R.TBIr_SbdmR-HXPoy?continueMode=true.

[41]E. Cachia, "GAPT Meeting 14th April", 2022. [Online]. Available: https://universityofmalta.zoom.us/rec/play/2WVc_N2Ck9rScOk3udYZQcyEA0NQxHycG9Sg43Sihhlv8znyDTK2zlSmPBd6vnD4DR8SwOj0O5ssqiZh.ZiUIUwqGDv-jM45n?continueMode=true.

[42]E. Cachia, "GAPT Meeting 29th April", 2022. [Online]. Available: https://universityofmalta.zoom.us/rec/play/8nnxLrIixDeo95h8MZL5UsXm1o5SRTUK-S0I7TJhptuSAv7ceJLIQndOkfDj_Pm3ybZ_0WdmW69dGIs0.c7k1z6Il5mqpVJnc?continueMode=true.

[43]"Plan-G", *Plan-G*. [Online]. Available: https://www.tasoftware.co.uk/.

[44]"ATC Communications and Radio Basics | Talking to Air Traffic Control 1", *Youtube.com*, 2018. [Online]. Available: https://www.youtube.com/watch?v=zBWVfz6GEgk.

[45]"Find My Plane for MSFS2020", *Findmyplane.live*. [Online]. Available: https://findmyplane.live/.

[46]"REST API (v3) - FsHub", *Fshub.io*. [Online]. Available: https://fshub.io/docs/platform/v3/rest-api.

[47]"FSX Flight Tracker", *Fsxtracker.com*. [Online]. Available: http://www.fsxtracker.com/.

[48]"X-Plane SDK", *Xsquawkbox.net*. [Online]. Available: http://www.xsquawkbox.net/xpsdk/mediawiki/Main_Page. [Accessed: 24- May- 2022].

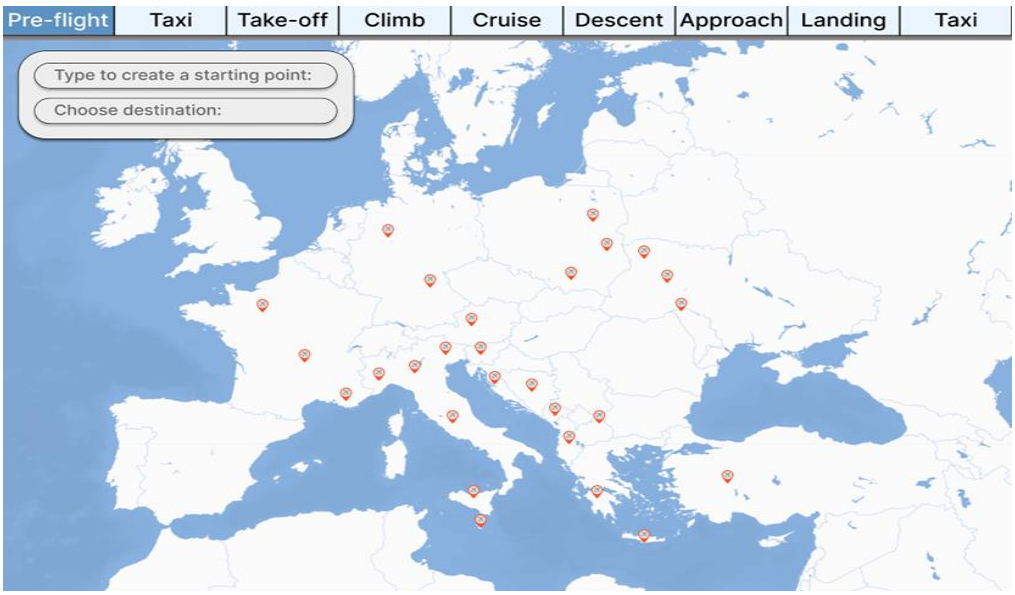
[49]"How to get location and heading data out of X-Plane", *X-Plane.Org Forum*, 2015. [Online]. Available: https://forums.x-plane.org/index.php?/forums/topic/90955-how-to-get-location-and-heading-data-out-of-x-plane/.

[50]"FlightGear forum • View topic - How to extract flight data?", *Forum.flightgear.org*, 2019. [Online]. Available: https://forum.flightgear.org/viewtopic.php?f=36&t=36256.

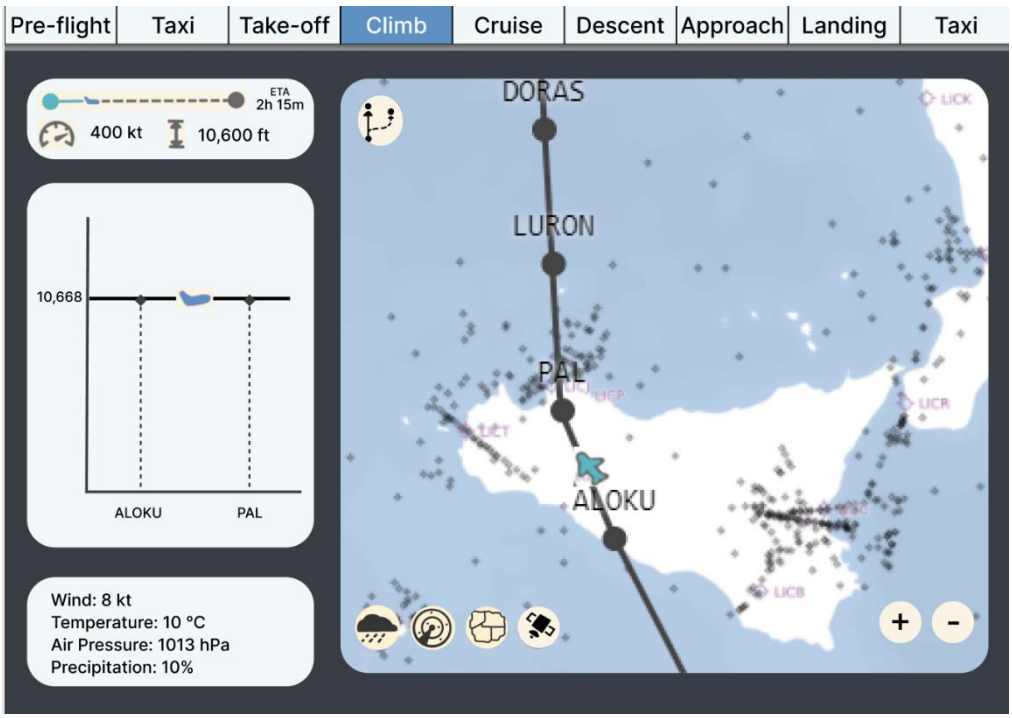
[51]"How to DECODE a METAR report (part 1) / Explained by CAPTAIN JOE", *Youtube.com*, 2020. [Online]. Available: https://www.youtube.com/watch?v=YWB2TrxmSfg.

9 – Appendix

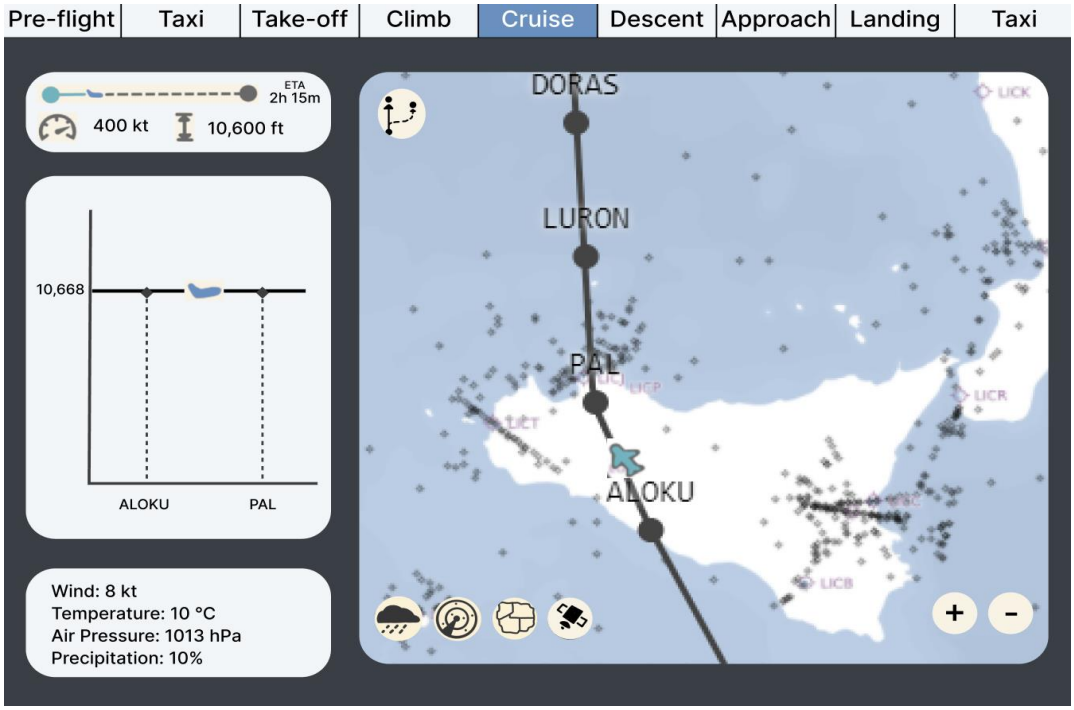
9.1 – Other Proposed Design pages for the Flight Visualisation System:



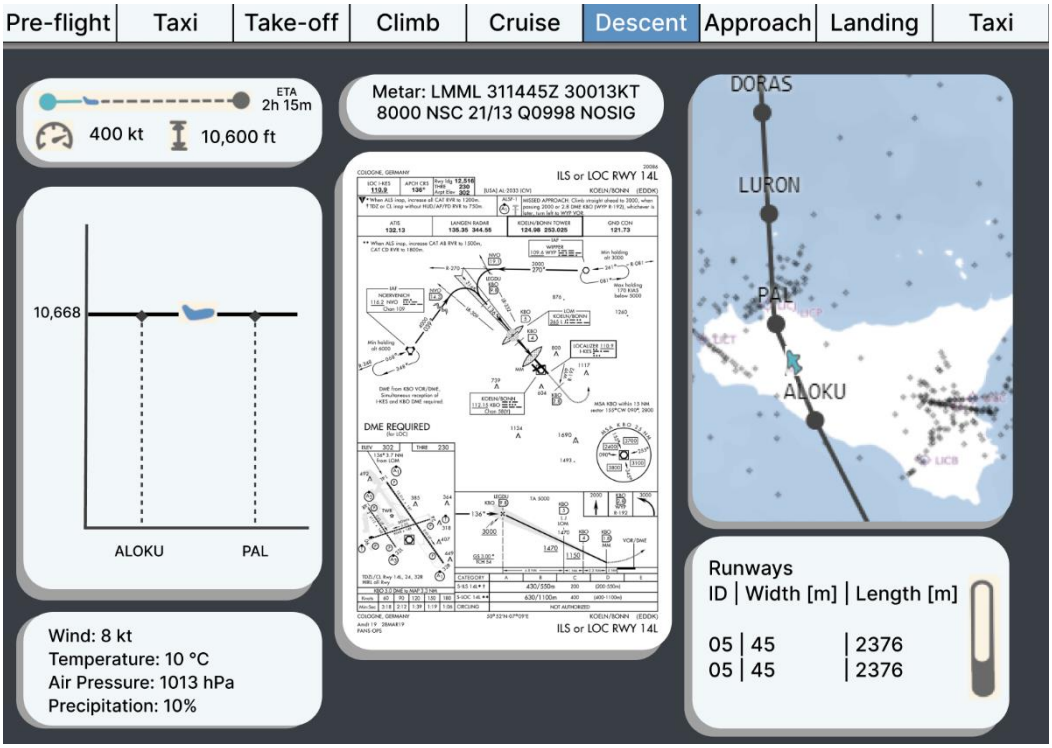
Proposed Design for Pre-flight phase start-up page



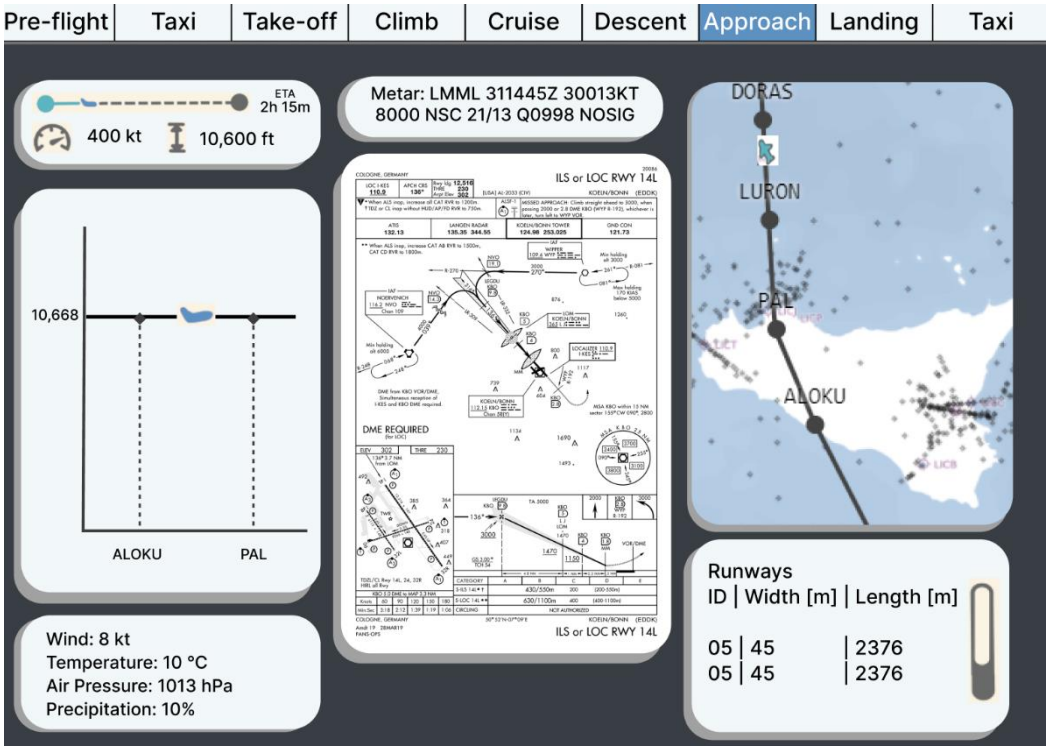
Proposed Design for the Climb phase



Proposed Design for Cruise phase

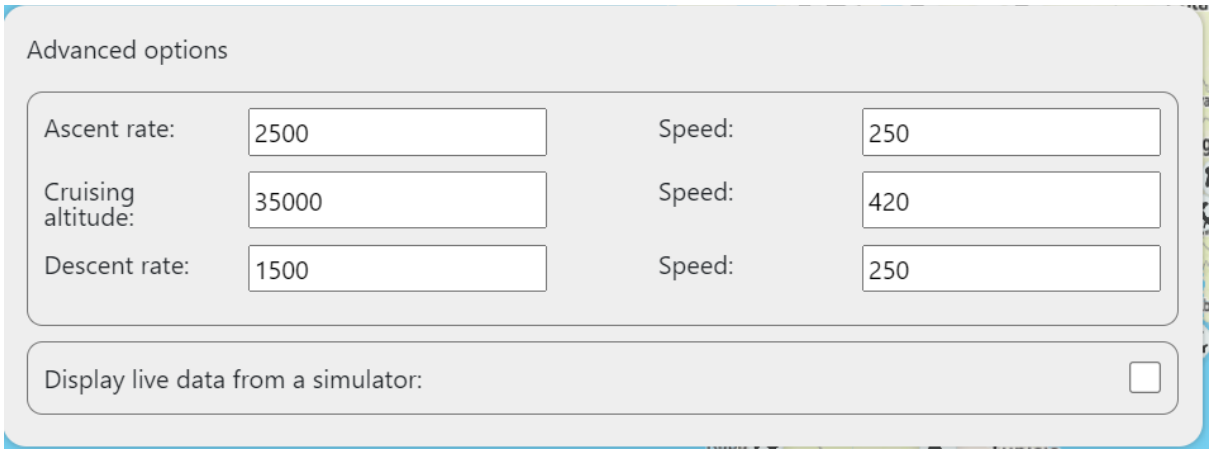


Proposed Design for Descent phase

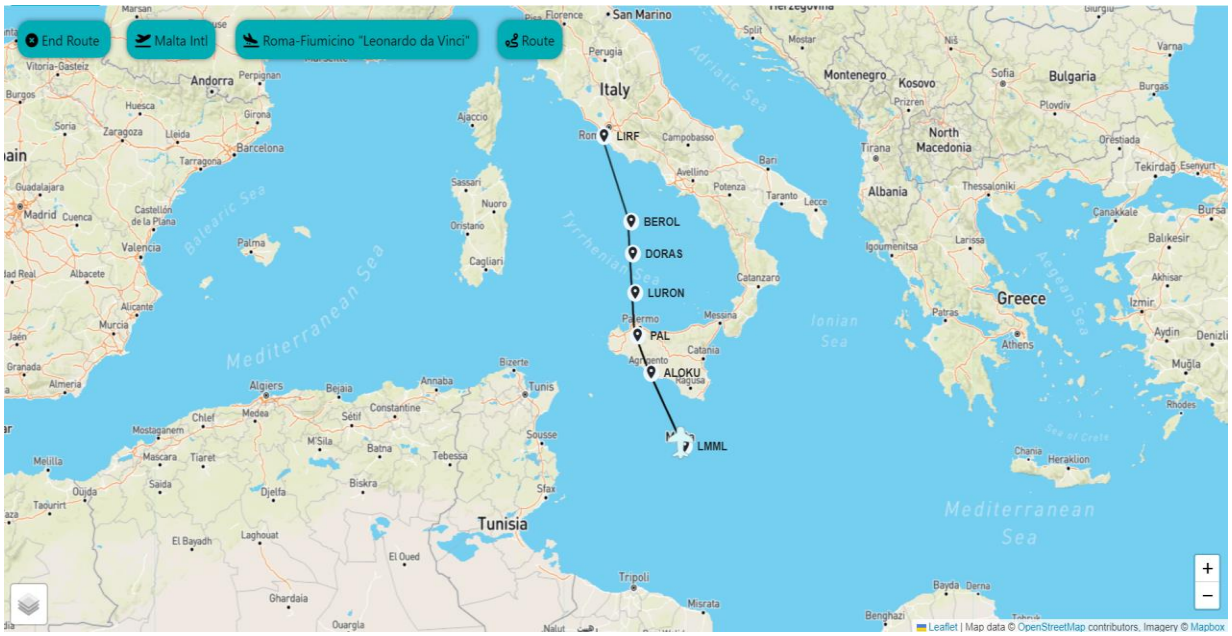


Proposed Design for Approach phase

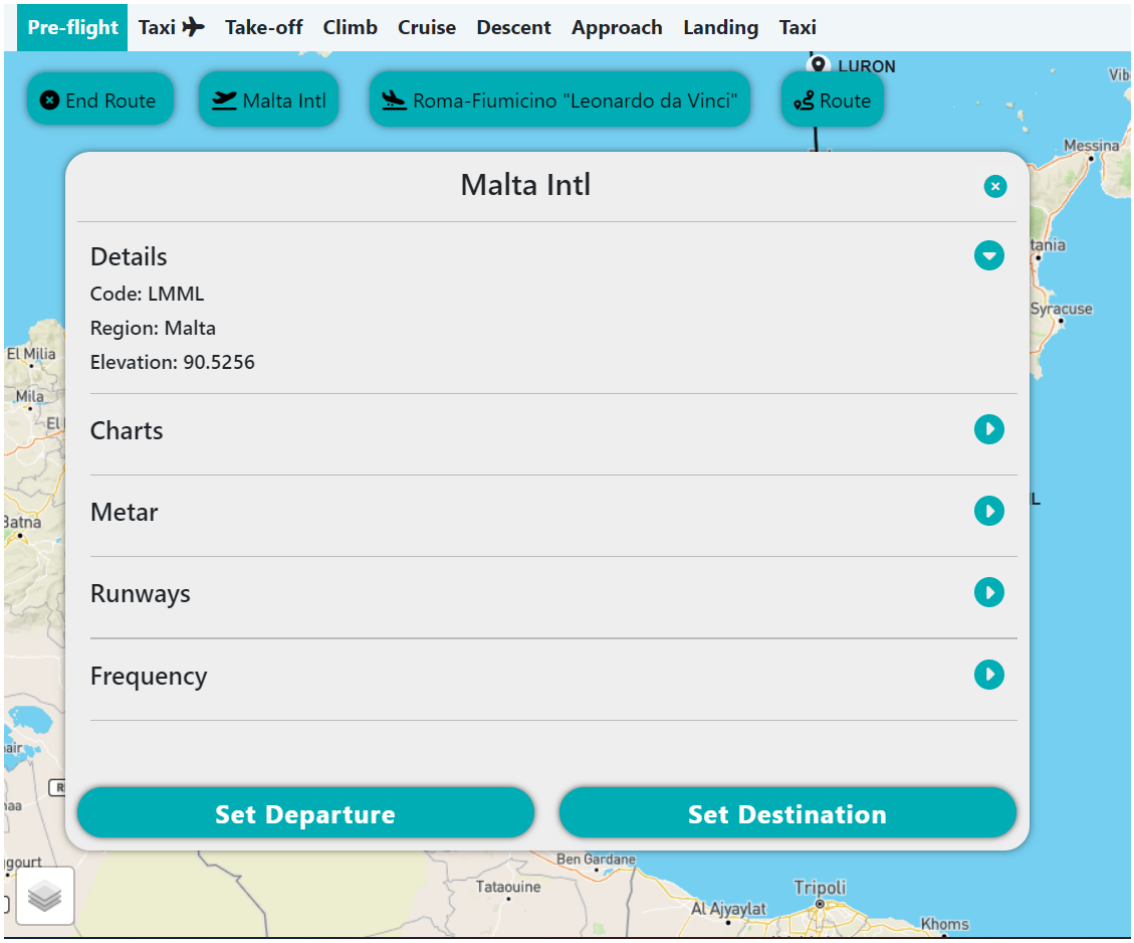
9.2 – Screen shots of Flight Visualisation System pages and features:



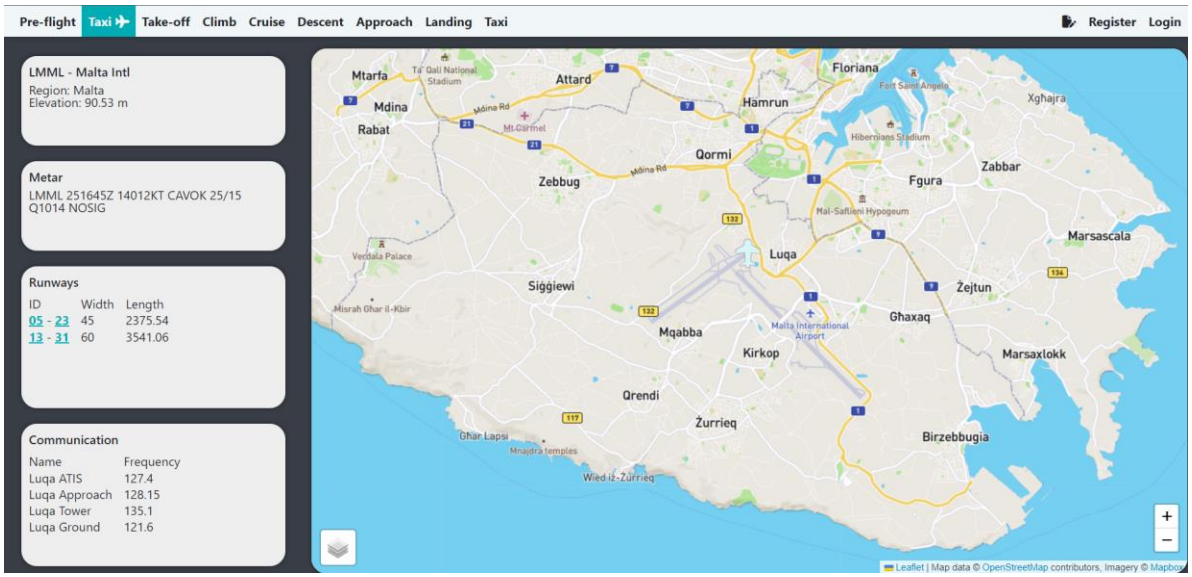
Features in the Settings menu



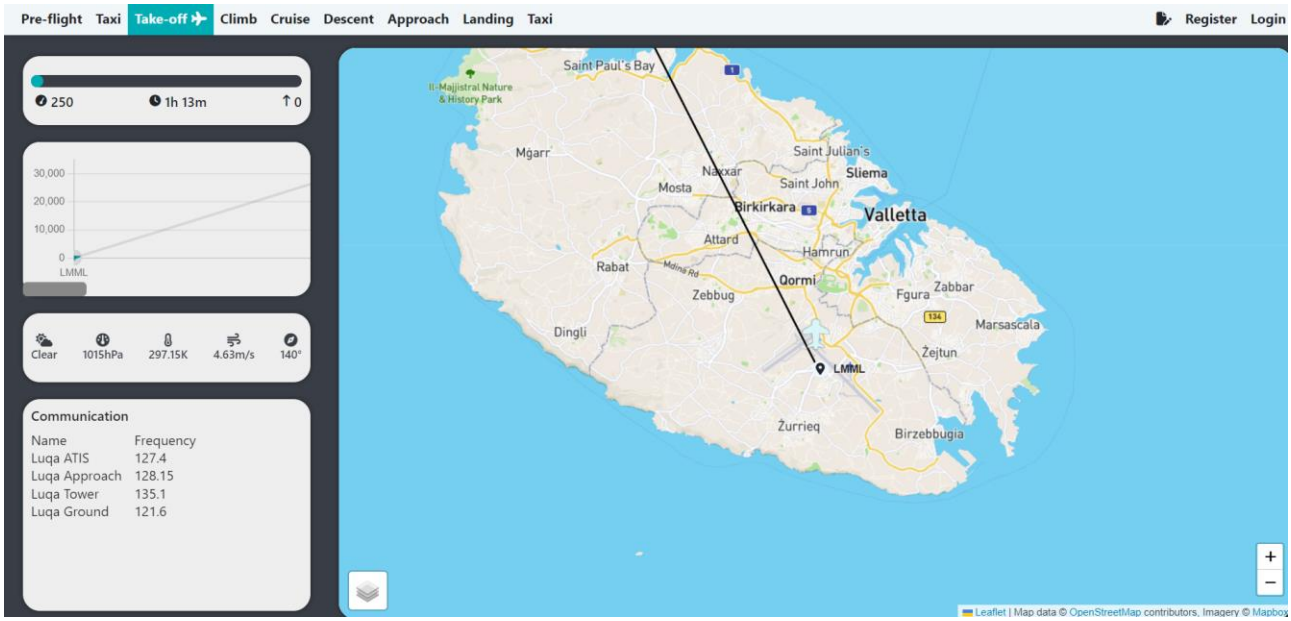
Preview of chosen Flight route in the Pre-flight phase



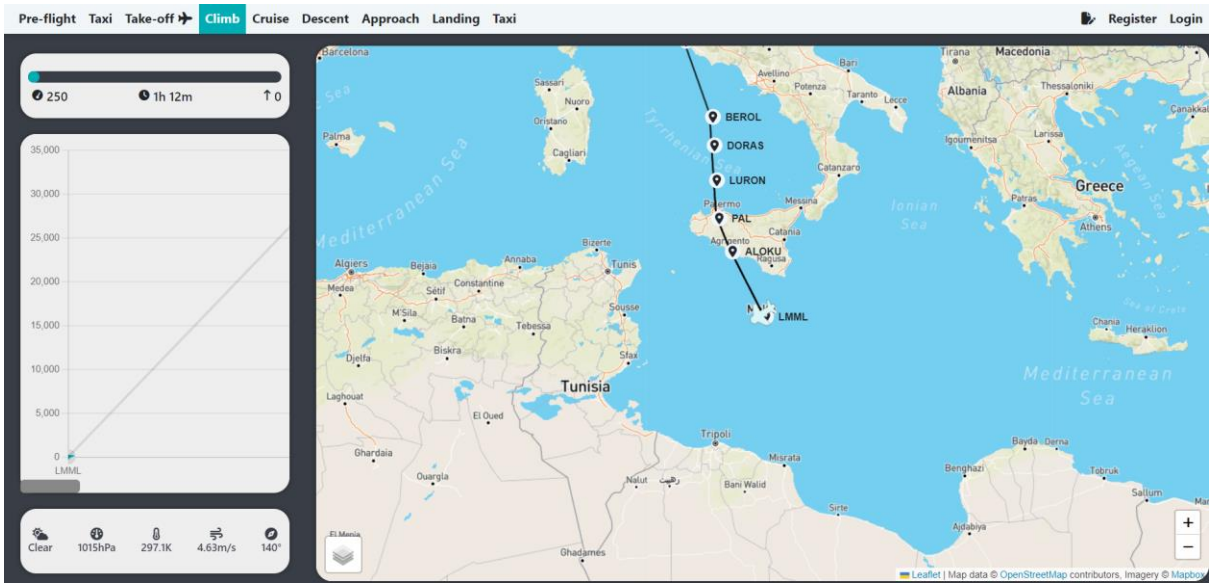
Available details of selected airport in Pre-flight phase



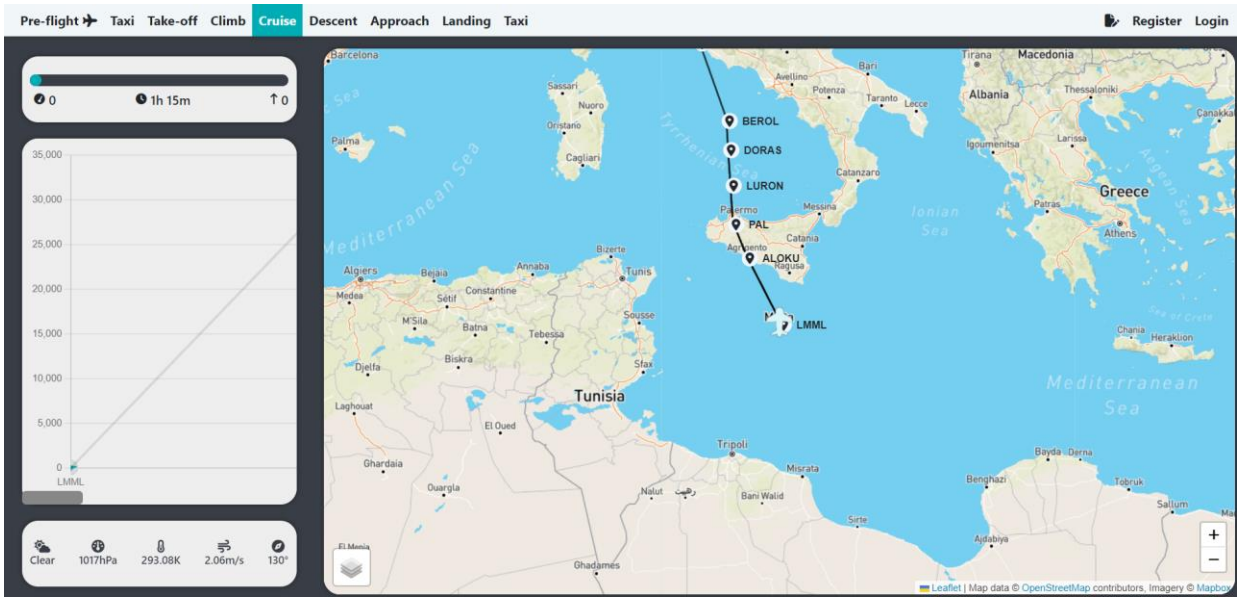
Taxi phase (departure) page



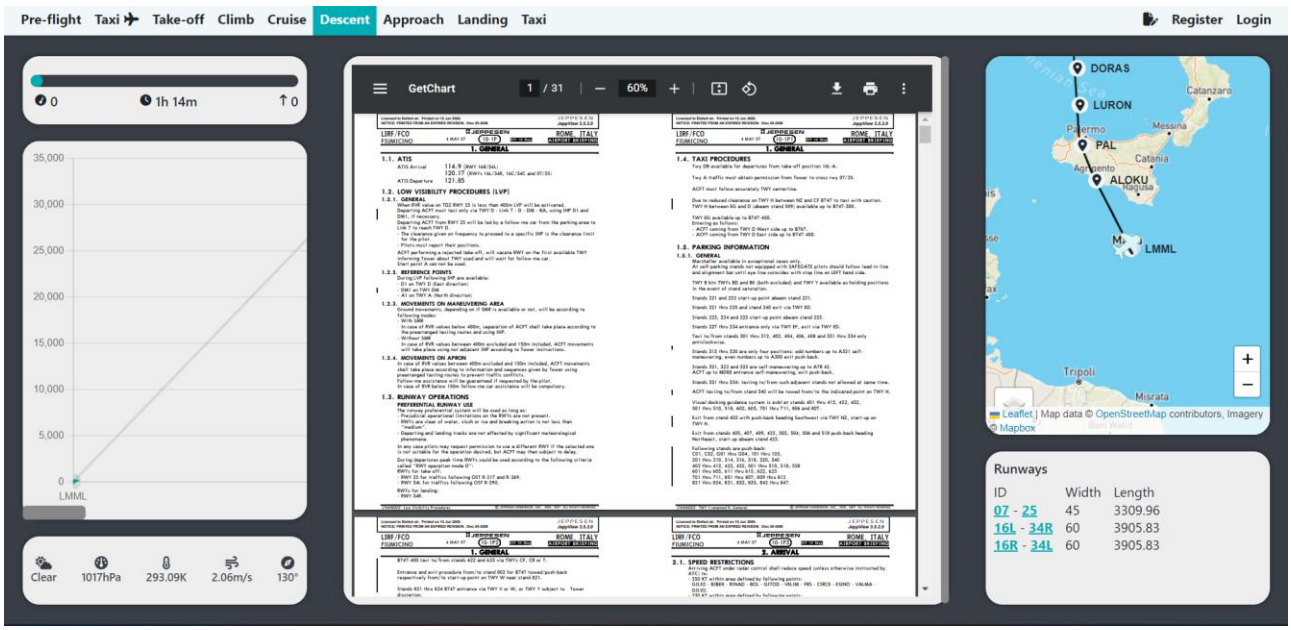
Take-off phase page



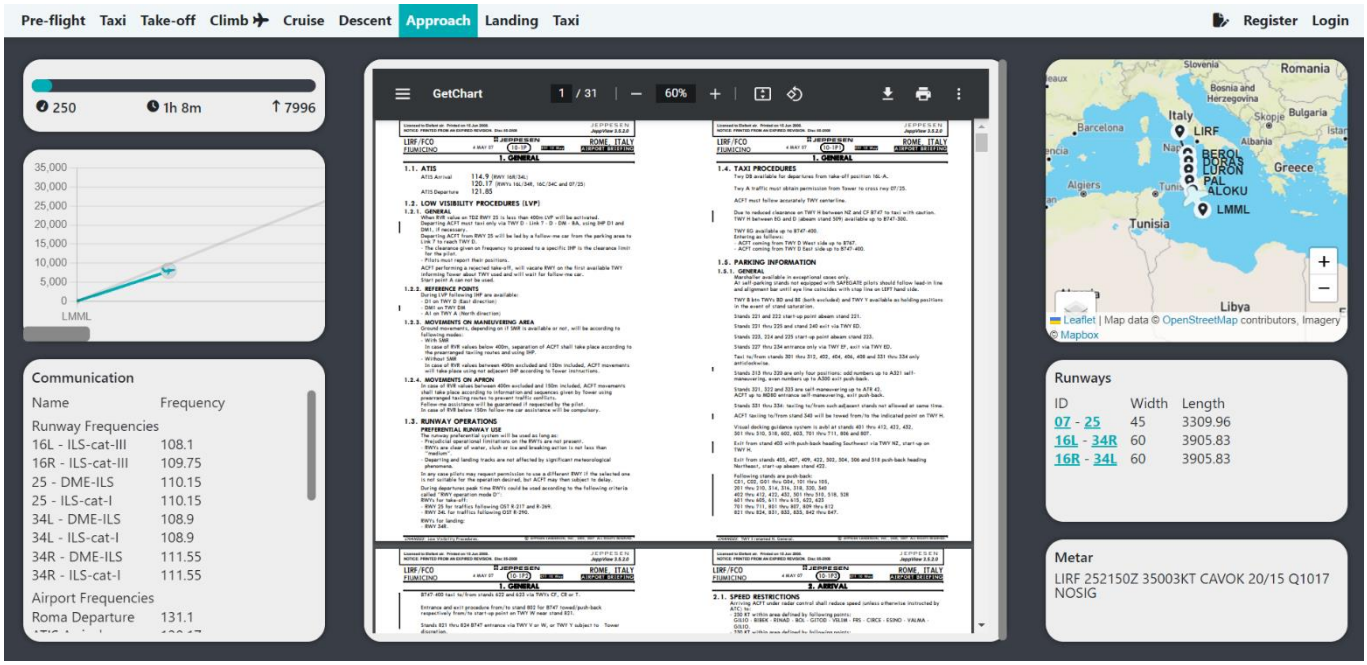
Climb phase page



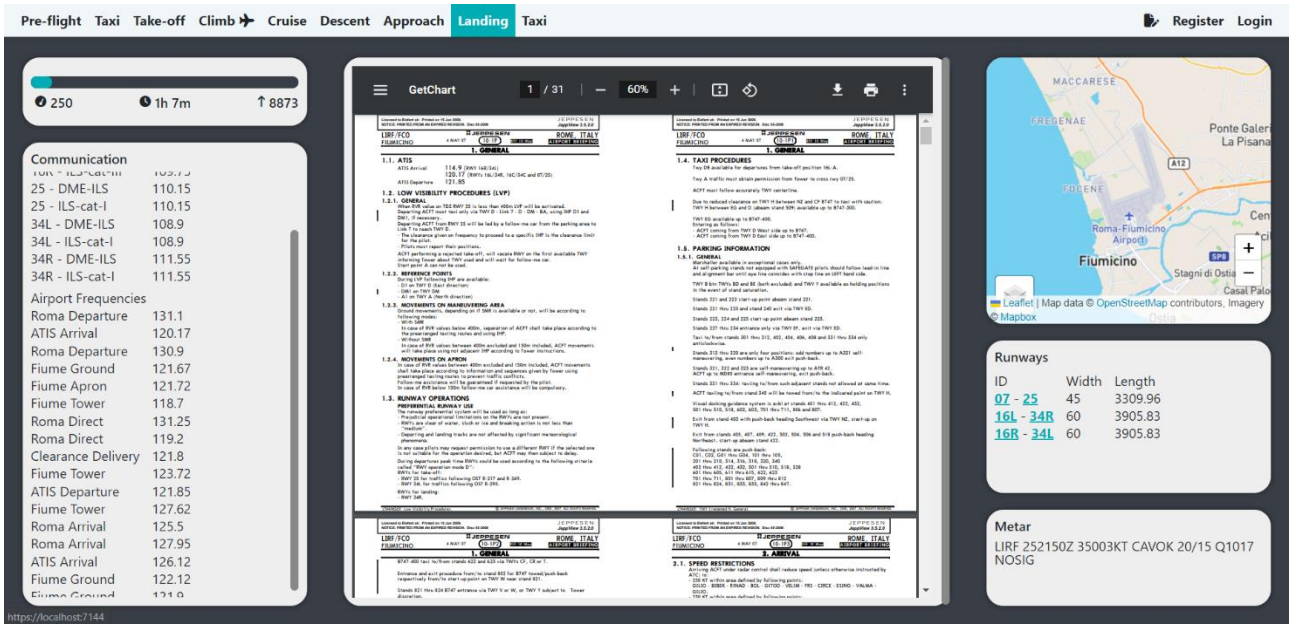
Cruise phase page



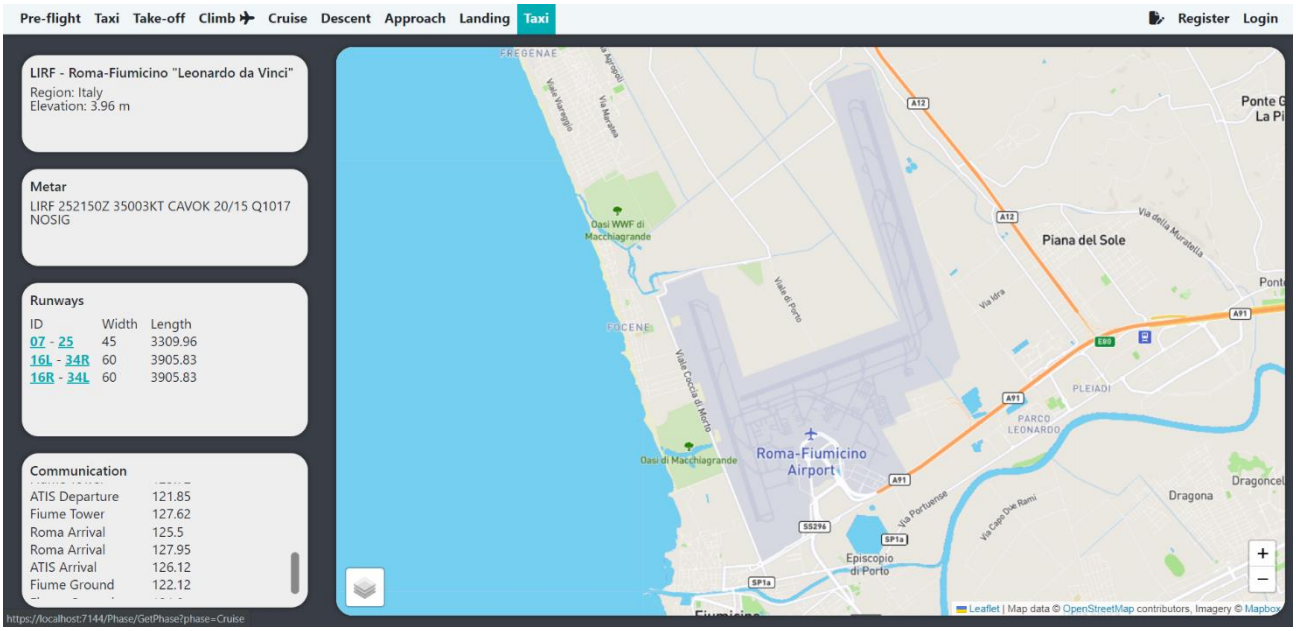
Descent phase page



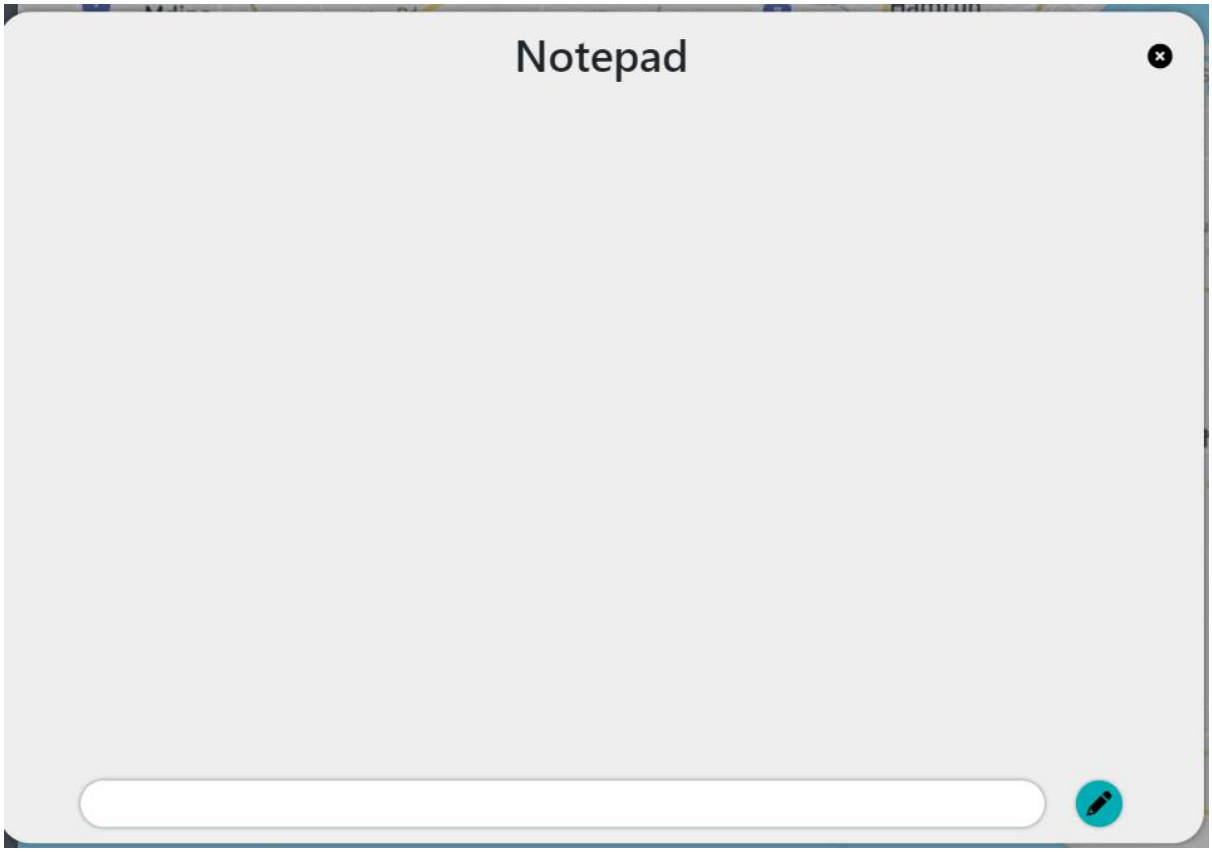
Approach phase page



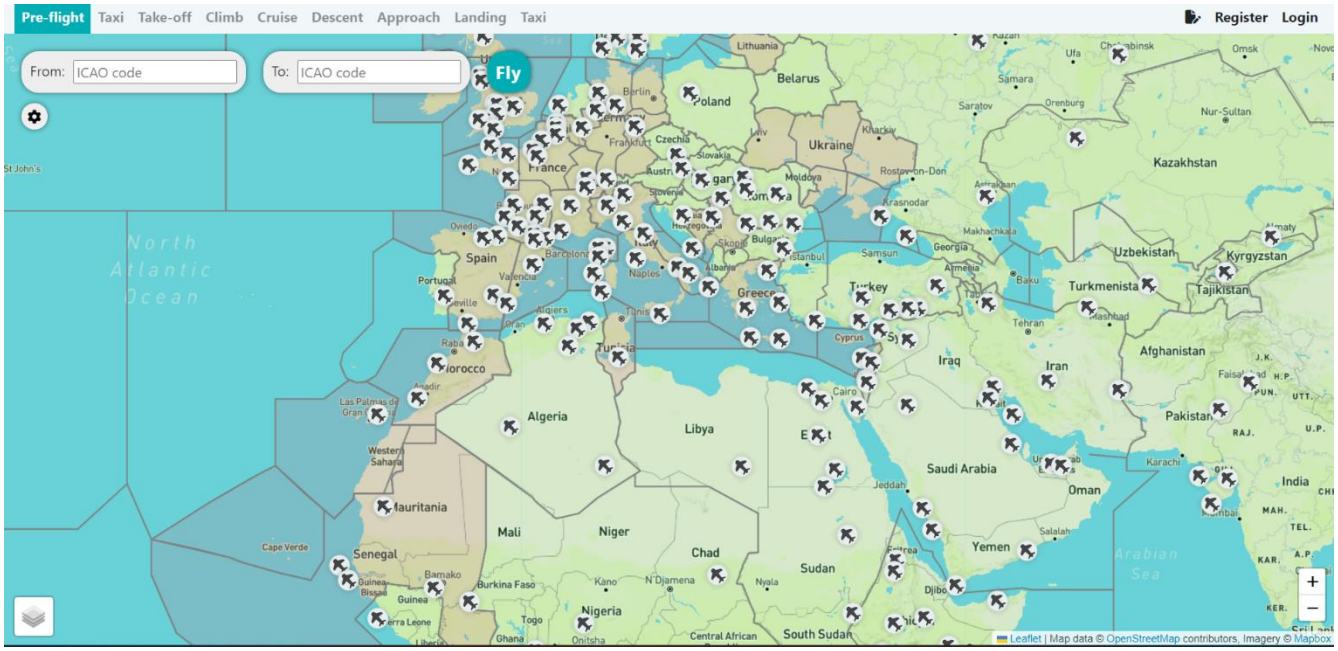
Landing phase page



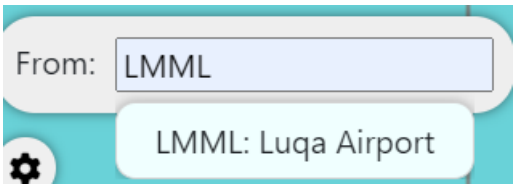
Taxi phase page



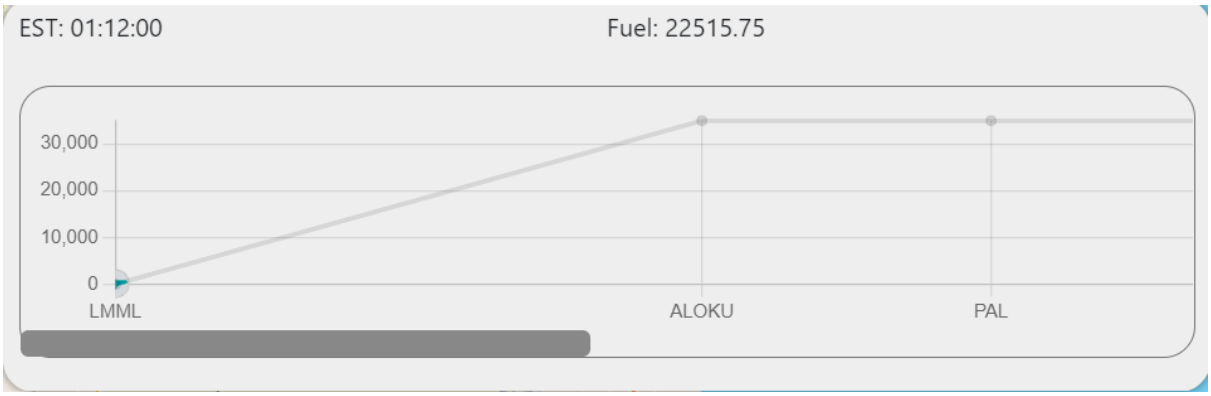
Notepad feature in the app



Enabling FIR feature on the map



Auto-fill feature in pre-flight page



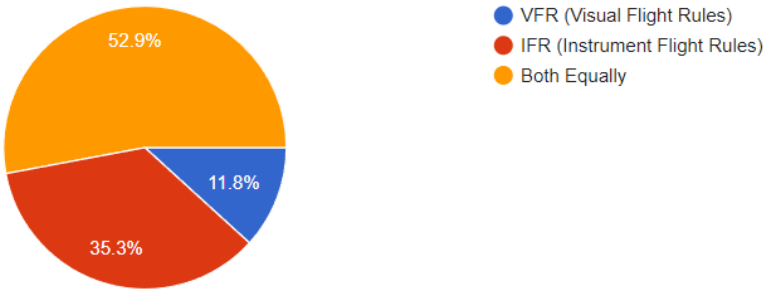
Estimated time of flight, required fuel and altitude components

9.3 – Survey Responses for Flight planning

Do you usually fly VFR or IFR?

Copy

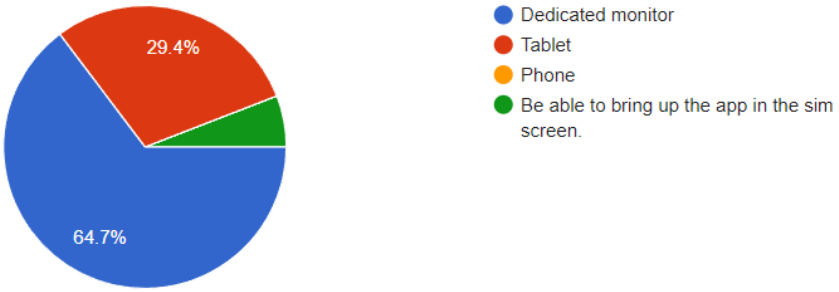
17 responses



How would you prefer to view an external application while simming?

Copy

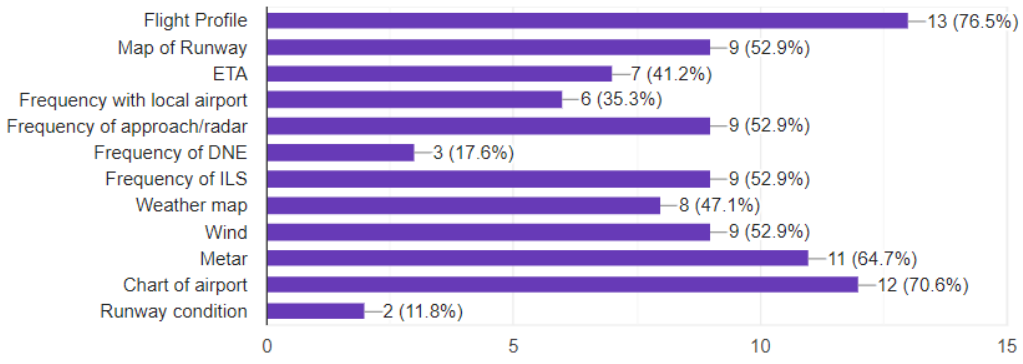
17 responses



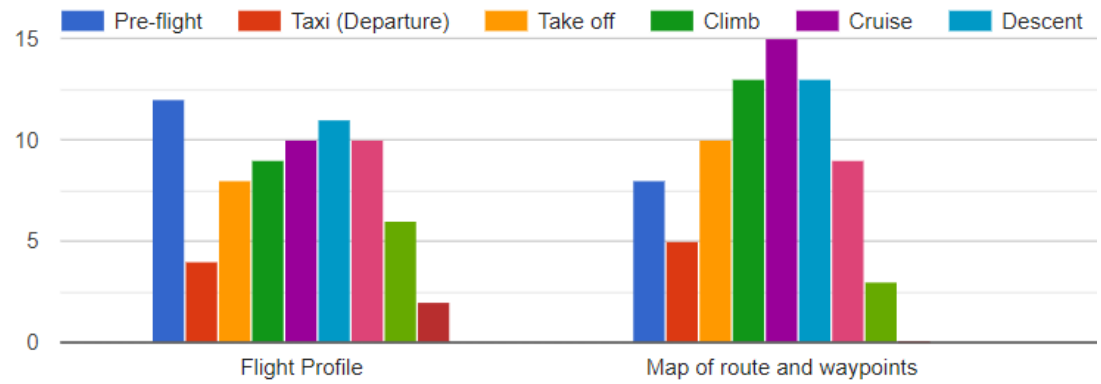
Which data would you need to see all the time and prominently during the simulation?

Copy

17 responses

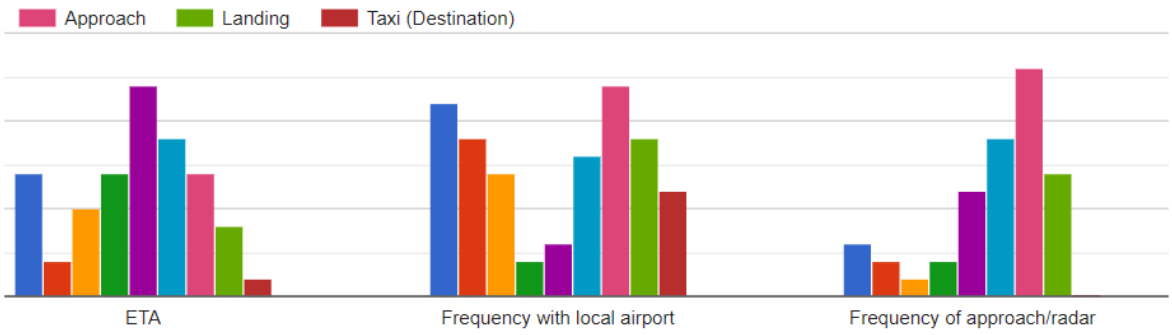


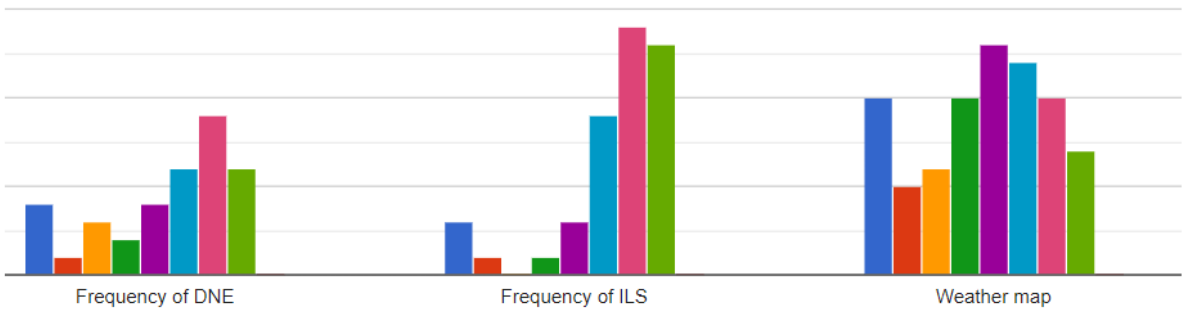
Which data would be useful to see in each phase?

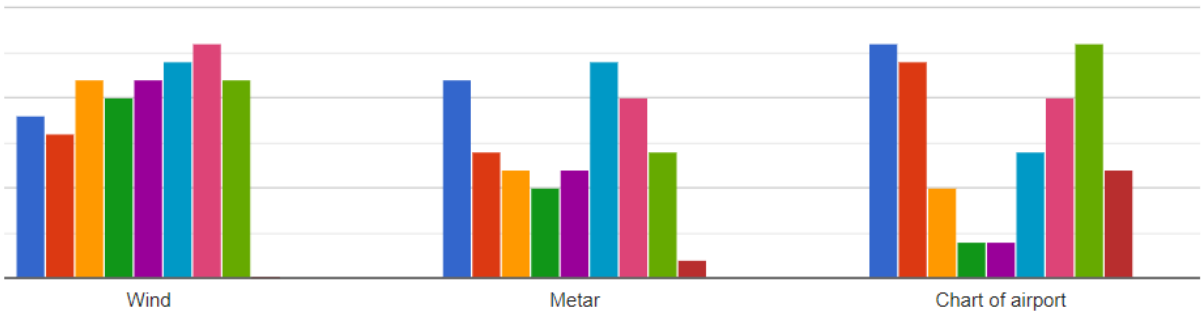


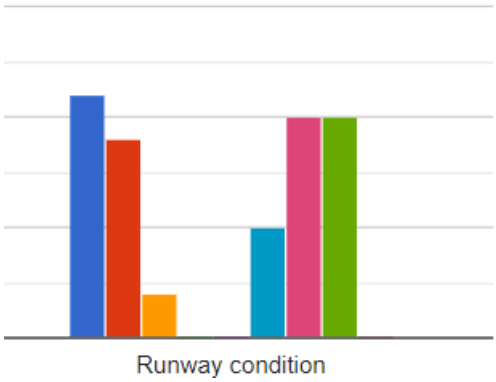
Which data would be useful to see in each phase?

Copy









What additional information would you require and during which phase(s)?

4 responses

NOTAM's - Preflight, Runway Info - Preflight

There might be an error - is it Frequency of DNE or Frequency of DME?

Thanks for allowing me to participate
Matthew Fenech

Checklists, preferably available throughout the flight.

It would be useful to give the user the chance to hide/unhide the above data in any phase of the flight

You seem to have my needs covered

9.4 – Meeting Logs

Group Assigned Practical Task
CIS2107

Meeting Log Sheet

The attendance log is to be shared with the supervisor(s) via Google Docs and must be updated on a weekly basis.

GAPT TITLE - Friendlier Flight Planning Visualization

SUPERVISOR - Ernest Cachia

STUDENTS - Kyle Agius **Andrea Fenech Cesareo, Neil Bugeja, Isaac Borg**

Date of Meeting	Meeting Outcomes	Deliverables for next meeting	Signatures
7/02/2022	- Introductory meeting	- Establish basic requirements - Methodology - Division of labour	AFC KA NB IB
			Supervisor EC
15/2/2022	- Planned Milestones - Determined methodology - Split the research tasks	- Feedback from supervisor, particularly regarding division of labour	AFC KA NB IB
			Supervisor

Date of Meeting	Meeting Outcomes	Deliverables for next meeting	Signatures
23/02/2022	- Discussed findings - Adjusted deadlines - Better way to split tasks	- Research different areas which we will be working on	KA IB AFC NB
			Supervisor EC
3/03/2022	- Outlining the questionnaire requirements	- Start working on the questionnaire	KA IB AFC NB
			Supervisor EC
05/03/2022	- Worked together on the questionnaire structure and questions	- Feedback from supervisor for the questionnaire	KA IB AFC NB
			Supervisor

10/03/2022	- Heard from supervisor about the questionnaire. Changes in the structure need to be made	- Work on the questionnaire and email the revised version to supervisor	KA IB AFC NB
			Supervisor EC

Date of Meeting	Meeting Outcomes	Deliverables for next meeting	Signatures
16/03/2022	- Finished questionnaire (post supervisor follow up via email) - Send questionnaire to forums and interest groups	- Await feedback from design	KA IB AFC NB
			Supervisor
18/03/2022	- Started showing the design sketches (Taxi) - positive feedback	- Do table of contents - Finish designing other stages	Students KA IB AFC
			Supervisor EC
21/03/2022	- Worked together on designs sketches for the pre-flight and	- Final responses from questionnaire will be used to decide which functions will be displayed in the simulator	Students KA IB AFC NB

	cruise sections of the simulator		Supervisor
25/03/2022	- Presented all the completed design sketches for the simulator and received feedback	- Revise and make changes to the designs - Start working on research write-up	KA IB AFC NB
			Supervisor EC

Date of Meeting	**Meeting Outcomes**	**Deliverables for next meeting**	**Signatures**
30/03/2022	- Presented first Figma prototype of the design (yellow bg)	- Decide on a colour scheme	Students KA IB AFC NB
			Supervisor EC
04/04/2022	- Colour scheme was decided and started to be updated in all flight stages on Figma	- Discuss way forward in easter holidays and when and how to start implementation	Students KA IB AFC NB
			Supervisor

08/04/2022	- Discussed way forward in easter holidays and set a date by which to start implementation once software to be used is decided	- Start documentation/report - Discuss software to be used	Students KA IB AFC NB
			Supervisor EC
14/04/2022	- Discussed what needs to be included in the introduction, research, requirements, design and methodology areas	- Introduction, research, requirements, design and methodology areas of report	Students KA IB AFC NB
			Supervisor EC

Date of Meeting	**Meeting Outcomes**	**Deliverables for next meeting**	**Signatures**
26/04/2022	- Met up to work on the implementation of the general design (cruise) - Worked on report areas discussed on 14/04/2022	- Continue working on implementation - Continue working on report	Students KA IB AFC NB
			Supervisor
27/04/2022	- Continued implementation of the general design (cruise) - Worked on report areas	- Await feedback for report areas	Students KA IB AFC *NB abroad
			Supervisor

29/04/2022	- We thoroughly went through the report documentation and took notes of adjustments which needed to be made	- Show the prototype to the supervisor	Students KA IB AFC NB
			Supervisor EC
03/05/2022	- Implemented pre-flight and taxi phases such that they are now functional. - Worked on report adjustments	- Show prototype of the website to the supervisor	Students KA IB AFC *NB abroad
			Supervisor
06/05/2022	- First in-person meeting with supervisor. We gave a demonstration of the prototype to the supervisor.	- Connection to simulator - Find frequencies and charts for airports	Students KA IB AFC *NB abroad
			Supervisor EC

Date of Meeting	**Meeting Outcomes**	**Deliverables for next meeting**	**Signatures**
10/05/2022	- Added airports to the Pre-flight and charts in the Descent. Approach and Landing phases. All 9 phases are now coherent and fully functional	- Connect website to simulator	Students KA IB AFC NB
			Supervisor

12/05/2022	- Online meeting between ourselves to connect the interface to MicroSoft Flight Simulator	-	Students KA IB AFC NB
			Supervisor
20/05/2022	- Final meeting before submission. Discussed finishing touches for website and documentation	- Implement final changes	Students KA IB NB *AFC illness
			Supervisor EC
TBD	- Mock presentation		Students KA IB AFC NB
			Supervisor EC