Product: I.A.N.
Team: SEVEN

Abstract

Over 4 billion passengers pass through airports every year, a number predicted to almost double by 2035.(1) Our goal is to make airports more accessible to disadvantaged groups by developing a personal assistant that helps users navigate airports to make travel easier and more enjoyable.

After the passenger has gone through security, they will be met by I.A.N., the Interactive Airport Navigator. This robot assistant will prompt passengers to scan their boarding pass and take a seat. Once sat down, passengers can receive real time information about their flight status, and can ask to be taken to their gate, or nearby facilities and amenities. Once I.A.N. is no longer needed, it will return to the hub.

1. Goal description

Finding your way through an airport can be difficult. There are usually several terminals each with multiple gates and the layouts are rarely intuitive. Airports can be overwhelming for those who rarely travel, and for those in foreign countries.

Our team proposes an airport assistant available to disadvantaged travelers. I.A.N. comes in the form of a chair which takes the passenger to their destination within the airport whilst providing useful information to the user as it goes.

1.1. Relevance of the system

A new study by the Transportation Research Board's Airport Cooperative Research Program shows that elderly passengers' ability to navigate airports is negatively affected due to the various aging-related issues like muscular or skeletal problems, mobility issues, and deteriorating sight or hearing.(2)

Moreover, for older individuals, anxieties and feelings of disconnect with the modern world are heightened in the fast-paced airport environment. The stress affecting anyone undertaking a major journey is felt more acutely by older people due to unfamiliarity with complex airport environments, unclear or confusing signs, and possibly language barriers.

I.A.N. aims to solve most of these problems by providing a stress-free way for elderly passengers to navigate an airport. Passengers will scan their boarding pass and I.A.N. will take them to their terminal or other points of interest. I.A.N. will be cheaper than paying for a private carer, and is always available for assistance.

We have taken inspiration from an existing project called SPENCER, a robot that helps KLM transfer passengers at Schiphol Airport, as well as KLM Care-E, a self-driving luggage trolley. They gave us the initial inspiration to take the airport assistant idea one step further and to develop I.A.N.

1.2. High-level description

When a passenger arrives at the I.A.N. hub, they should be able to scan their boarding pass on I.A.N. (either by themselves or with the assistance of airport staff). I.A.N. should then acknowledge the passenger's flight number and status, and would then prompt the passenger to sit down. In addition, the airport staff at the I.A.N. hub could help to load the passenger's luggage into I.A.N.'s dedicated luggage compartment.

When the robot recognises that the passenger has sat down (by means of a weight sensor), it waits for a few moments until it begins to move. Once the robot arrives at the passenger's destination, it should remind the passenger to pick up their luggage. After the passenger retrieves their luggage, I.A.N. should wait for a few seconds, wishing the passenger a safe flight, before heading back to the I.A.N. hub.

To extract the possible features of the system we mainly used user stories. We tried to think of users with different kinds of impairments (like mobility, hearing, or vision issues) to understand what the user might require from the robot. However, with that being said, and due to time and resource limitations, our main target group for the purposes of this course should be those who *can* get around airports but still find it difficult to do so.

The high-level functionalities are summarised in the following list:

- There will be I.A.N.s available to use at the I.A.N. Hub.
- The passenger can scan their boarding pass so I.A.N. knows what flight they are taking.
- They can listen to or read flight information, current flight status, and gate number to stay up to date on their journey.
- They can put their luggage in I.A.N.'s luggage compartment.
- They should know when I.A.N. is ready to start moving so they can take a seat, and I.A.N. will give them enough time to make themselves comfortable.

- They can be taken to their gate or other points of interest to avoid the difficulty and stress of finding the way themselves.

- I.A.N. will avoid obstacles on their journey.

- They will be reminded to take their luggage so they don't forget it, and I.A.N. won't leave until they've have done so.

- I.A.N. will be friendly and kind.

1.3. Target audience

Although I.A.N. has the potential to help many different groups of people, we decided to focus on aiding the elderly. I.A.N.'s main objective is to guide passengers around the airport, which is especially challenging to those with mobility issues in cases where they're expected to walk long distances. However, this is not an issue exclusive to older people as this could be a concern for pregnant women and those with injuries and illnesses.

A straightforward and simple user interface would make I.A.N. accessible to the most technologically inexperienced passengers. Further features helping other target audiences, such as those with visual or hearing impairments, include flight information text-to-speech, voice commands, and possibly physical assistance getting into the chair. Furthermore, I.A.N. could communicate in multiple languages to help those travelling abroad. However, given our limited time, resources and experience, we have decided to focus on one target audience, this being the elderly.

2. Task planning

2.1. Milestones

The following are the specific milestones needed to accomplish each of the user stories given earlier. In the following subsection, these milestones are broken down further into defined technical tasks.

1. First Demo Milestones
   (a) The physical environment for the robot is set up.
   (b) The robot can move from any point in the airport to a predefined point, such as a gate.
   (c) The robot can return to its starting position after it has reached the gate.
   (d) The robot can scan a boarding pass in the form of a QR code.

2. Second Demo Milestones
   (a) The robot can move to more than one location in the airport.
   (b) The robot avoids static obstacles while moving.
   (c) The database/server for flight information is set up.
   (d) The robot can retrieve real-time information from the database, such as flight number, gate/terminal and flight status.

3. Third Demo Milestones
   (a) The robot can take the user to restaurants, toilets and other points of interest.
   (b) The robot avoids dynamic obstacles while moving.
   (c) The robot can present real-time flight information on a screen and via text-to-speech.

4. Final Demo Milestones
   (a) The user seat is installed.
   (b) The robot knows when the seat is occupied using a pressure sensor.
   (c) The luggage compartment is installed.
   (d) The robot alerts the user when the luggage compartment is occupied and the passenger seat is not.
   (e) The visible parts of the robot are made aesthetically pleasing.

2.2. Task decomposition

Each milestone outlined earlier is decomposed into one or more tasks in Table 1. In addition, a Gantt chart has been included below in Figure 1 to help visualise the timeline of the semester. The dependencies between those tasks are outlined only for the tasks that are in the same sprint in the run up to a certain demonstration. This was done to enhance the readability of the Gantt chart.

Although we hope to strike a balance between being passionate about our goals and maintaining realistic expectations, we accept that not all of the tasks in the Gantt chart may be met. If we run into challenges, the Gantt chart will be revised.

2.3. Resource distribution

2.3.1. Time Management

SDP is a 20 credit course which means each team member is expected to spend 200 hours on the project. The expected breakdown of this time is shown in Figure 2.

- Research & Development (144 hours) - building, programming, testing, and debugging the system.

- Pitches, Demos, Trade Fair (12 hours) - the initial pitch at the start of the year, the four demos, and the investor trade fair.

- Meetings & Planning (20 hours) - the weekly meeting with our mentor, as well as regular scrums and team meetings.

- Writing Reports & Pitches (15 hours) - writing the initial group pitch, the project plan, the four demo reports, the user guide, and the individual report.

- Workshops (5 hours) - attending workshops and Q&A sessions.
- Talking with Experts & Technicians (4 hours) - getting advice from experts and help from technicians.

2.3.2. Equipment and Physical Resources

The main component of our product is a Turtlebot 3 Waffle Pi, this comes equipped with sensors and enough processing power for our needs. We will be using a combination of gyroscope, accelerometer, cameras and laser distance sensors, which are already on the robot, for localization purposes. A camera module will also be used to scan the users' boarding pass. Additional sensors such as pressure sensors will be used to detect when the user has stopped using the chair and to make sure that they collect their luggage. In order to make our device accessible, we will be using a touch display and an Amazon Alexa to provide information to the user but also to take instructions.

The final design should be chair-like, with the display built into the armrest. In order to build this, we plan to use a combination of MDF sheets and Lego.

2.4. Risk assessment

Some of the major technical risks that the team may face are difficulties in the use of hardware due to the team's lack of experience in the area. Our plan to mitigate this (as discussed) is to create a sub-team devoted entirely to the hardware aspect of our project. We believe that having 5 people working on this early will minimise the impact of our inexperience.

The integration of components into the complete system will prove to be challenging. For this reason we plan to integrate as often and early as possible. We also plan to use the time before the final demonstration to polish the final product and fully integrate all remaining sub systems. We know it is essential for the sub-teams to maintain clear communication in order to ease this process.

Another important factor that could hinder the project's progress is the absence of team members. Our splitting the team into sub-teams with multiple members ensures that no one team member has knowledge that no other team member possesses. We also have technical managers/coordinators to make sure the whole team is kept up to date on the project's progress. This way, we minimise the impact of any team member's absence. See Table 2.

Finally, there is a risk of failing to reach our milestones. This course aims to provide us with a real-life work experience where agile methodology would be used and we realise it's possible some of our goals will be put aside during later stages of development when we have a better understanding of technical limitations, and time and equipment constraints. We acknowledge that risk and our contingency plan is to follow the agile development value of "responding to change over following a plan" and will change our goals to match our constraints.

3. Group organisation

3.1. Structure

Our team employs a horizontal structure. We have three teams: Software, Hardware and Environment, each of which have unique responsibilities. The Environment team is considerably smaller than the other teams as setting up and maintaining the environment should require less work than developing the software and hardware of the robot. The team assignments were based on a combination of preference and competence. Consequently, each member can work on aspects they find interesting while we also make use of current strengths and expertise within the group. The team assignments are shown in Table 3.

To ensure effective communication we have assigned a coordinator to the software and hardware teams. Their role is to organise the tasks their team must complete by splitting tasks among team members. The coordinator should also facilitate communication between sub-teams and ensure everyone is working effectively.

Furthermore, we have team members assigned to two different teams to mitigate any risks of miscommunication. These team members were selected because of their communication skills and will take on the role of passing information between the respective sub-teams. This takes the pressure off the team coordinators also having to carry out this role and has the added benefit that each team has an understanding of other parts of the system.

Lastly, we want our teams to be flexible. Therefore, there is room for movements between teams/sub-teams which could occur due to an unexpected workload. Also, developers are encouraged to work on any feature that they find interesting which could also cause shifts. Any such changes would be discussed with the entire team.

3.2. Management

Given our flat structure, our group does not have a designated manager. We reasoned that taking on the role of manager is a big responsibility and requires that an individual has a large skill set beyond technical skills. Since we did not have an obvious manager within our group we decided that it would be unfair to appoint one. We also wanted to focus on effective team work and thought that one individual having more authority than others would be a detriment.

Despite this, the roles carried out by a manager are essential to a well functioning team. Therefore, we decided to split up typical managerial roles by having specific members carry out organisational and communicative roles within and between our teams. To stay on track, we plan to hold regular scrums to stay up to date on individuals' progress and ensure everyone is clear on their current goals.

3.3. Meetings

Every morning we have the group members give an informal update on their progress in order to keep the whole team up to date. This update is given in written form on Nuclino and includes any new issues that have emerged.

We also have a scheduled weekly meeting with our mentor on Friday afternoons, in which everyone is expected to attend. There will also be meetings after each demonstration to discuss the client feedback, identify any new issues and plan steps to mitigate them.

Each sub-team is responsible for arranging their own meetings according to their schedule and needs. Additional team meetings follow a drop-in approach and are conducted as needed.

3.4. Communication and Tools

For communication we use Slack to directly message the group or any particular team member about a project update or issue. We use Nuclino as our main platform for notes and drafts. We also use Trello for project management and task allocation as it allows assignment of specific tasks to each team member via To-Do boards. It also provides an easy way to update the team on what tasks are still under development and which ones are already finished. A private GitHub repository for code version control has been set up for the project. We use GitHub as everyone has experience with it. Lastly, we use TeamGantt to create and update the Gantt chart.

References

[1] The World of Air Transport in 2018,
https://www.icao.int/annual-report-2018/Pages/the-world-of-air-transport-in-2018.aspx

[2] Airport Cooperative Research Program: Impacts of Aging Travelers on Airports,
http://3rxg9qea18zhtl6s2u8jammft-wpengine.netdna-ssl.com

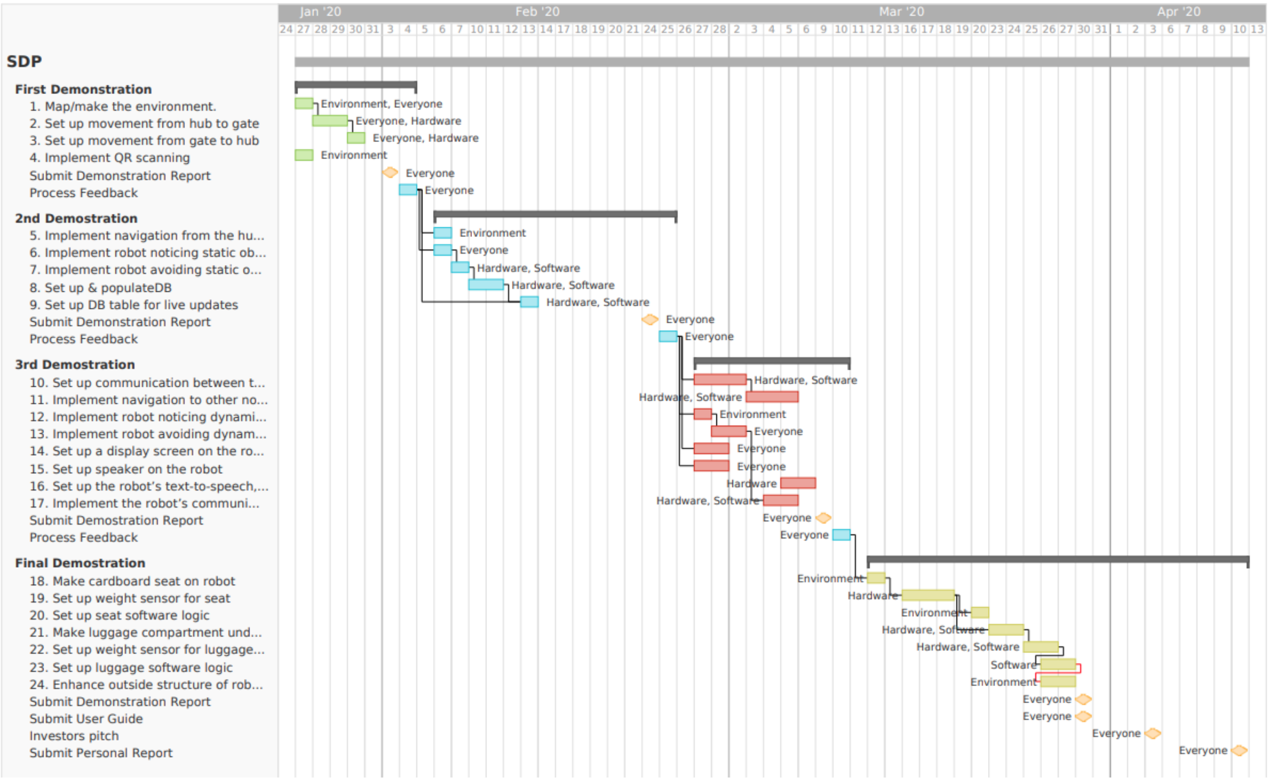

Figure 1. The Gantt chart for the project.

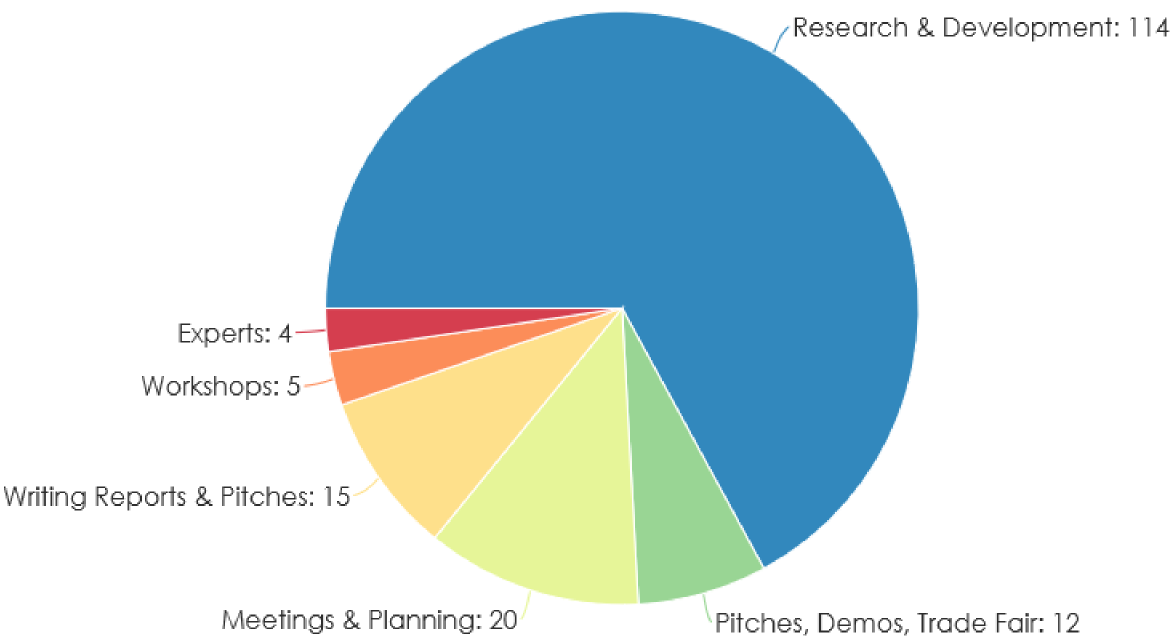

Figure 2. Projection of hours allocated to categories.

Task Name	Milestone	Estimated time	Dependency	Rough description
1. Map and make the environment	1.a.	0.5-1 Days	-	Set up the environment the robot will move in
2. Set up robot navigation from hub to gate	1.b.	2-3 Days	Task 1	Implement functionality for the robot moving to a gate
3. Set up robot navigation from gate to hub	1.c.	2-3 Days	Task 2	Implement functionality for the robot moving back to the hub
4. Implement QR scanning	1.d.	1 Day	-	Boarding passes in the form of QR codes can be scanned
5. Implement navigation from the hub to multiple gates and back	2.a.	1 Day	Task 3	-
6. Implement robot noticing static obstacles	2.b.	2-3 Days	Task 1	-
7. Implement robot avoiding static obstacles	2.b.	2-3 Days	Task 6	-
8. Set up & populate DB	2.c.	0.5-1 Days	-	Set up DB containing multiple gates and flight information, with QR codes matching these entries in the DB
9. Set up DB table for live updates	2.d.	1 Day	Task 8,5	Update table as bot moves
10. Set up communication between the robot & the DB	2.d.	1 Day	Task 8	Have the bot able to retrieve info from the DB
11. Implement navigation to other notable locations in DB	3.a.	0.5-1 Days	Task 5,10	Set up other airport locations e.g. toilets and navigation to them
12. Implement robot noticing dynamic obstacles	3.b.	3-4 Days	Task 7	-
13. Implement robot avoiding dynamic obstacles	3.b.	3-4 Days	Task 8	-
14. Set up a display screen on the robot	3.c.	1 Day	-	Set up the screen ready to display info
15. Set up speaker on the robot	3.c.	1 Day	-	Integrate a speaker on the robot
16. Set up the robot's text-to-speech, testing on predefined words	3.c.	2-3 Days	Task 15	-
17. Implement the robot's communication of live updates from DB	3.c.	1-2 Days	Task 10, 14, 16	The updates displayed on screen and read by text-to-speech
18. Make cardboard seat on robot	4.a.	0.5-1 Days	-	Create a mock seat on the robot
19. Set up weight sensor for seat	4.b.	1-2 Days	Task 18	-
20. Set up seat software logic	4.b.	1-2 Days	Task 19	The robot knows when the seat is occupied
21. Make luggage compartment under seat with cardboard	4.c.	0.5-1 Days	Task 19	-
22. Set up weight sensor for luggage compartment	4.c.	1-2 Days	Task 21	-
23. Set up luggage software logic	4.d.	1-2 Days	Task 22	Set up logic to avoid luggage being left behind
24. Enhance outside structure of robot	4.e.	1-2 Days	Task 23	Clean up the outside of the robot to make more aesthetically pleasing

Risk	Plan
Team member being absent	Assign people in multiple teams. Eliminate the chance that a team member has unique knowledge about the system.
Lack of hardware experience	Allocate a hardware-oriented team to get familiar with the hardware as early as possible.
Integration of sub-systems	Communication between sub-teams and team members should allow for a smooth integration of sub-systems.
Critical items acquisition	Outline and order items that are critical as soon as possible.
Support staff might be unavailable	Following the schedule will minimise the possibility of needing last minute support.
Unrealistic features	Mitigated by careful consideration of milestones and use of agile development methodology to respond to change.
Legal/Ethical constraints	Research on legalities and ethical implications of our system will minimise this risk.

Table 2. Risk Assessment Summary

Member	Team	Reason
Chongli Shen	Hardware/Software	Desire to learn a new skill/Experience in developing applications.
Ethan	Hardware	Desire to learn a new skill.
Maria	Hardware	Desire to learn a new skill.
Daragh	Hardware	Engineering and building experience.
Matt	Software	Data analysis experience.
Erodotos	Software Coordinator	Knowledge in software engineering.
Yuexiang	Software	ML development experience.
Yussef	Software/Environment building	Data science background and software engineering experience.
Eloise	Hardware/Environment building	Desire to learn a new skill.
Ion	Hardware Coordinator	Experience with electronics.

Table 3. Team Assignments

Tool	Purpose
Slack	Main Communications HUB
Nuclino	Meeting Notes, Drafts
Github	Code Version Control
Trello	Project Management, Task Allocation and Updates
TeamGantt	Gantt chart design

Table 4. Tools used for System Design Project