Experiment 6: To develop Software Requirement Specification(SRS) for taken up project

***Introduction***

1. **Background**

The project is titled as airport self check-in system. The main idea behind the project is to save time of passengers at the check in counters in the airport. With the advancement of technology, especially in the fields of biometrics, machine learning, and user interfaces, self-check-in systems have become more feasible and user-friendly. This has enabled airports to offer self-service options to passengers with greater confidence in security and reliability. Offering self-check-in options provides passengers with greater flexibility and control over their travel experience. They can choose their seats, print boarding passes, and check baggage independently, reducing reliance on airline staff and enhancing overall satisfaction.

1. **Overall Description**

Self-check-in airport systems are automated procedures that let travellers check in for their flights without help from airline employees. Usually located in airports, these systems are designed to expedite the check-in procedure, shorten wait times, and improve the general traveller experience.

Airlines often provide a combination of digital kiosks, smartphone apps, and web platforms for the self-check-in procedure. These technologies allow travellers to pick their seats, print boarding tickets, check in and, in certain situations, tag their own luggage.

Self-check-in systems are being used due to a number of reasons, including as the desire to boost productivity, cut expenses, and accommodate passengers' shifting preferences for digital services. Utilising technological innovations like biometrics, machine learning, Leveraging user interfaces to offer a safe and streamlined check-in procedure.

Airports and airlines hope to maximise resources and improve security measures while simultaneously giving travellers more choice over their travel experience by providing self-service alternatives. All things considered, airport self-check-in technologies are a big step towards updating aviation procedures and satisfying the changing demands of passengers.

**c. Environmental Characteristics**

**i. Hardware**

Airport self-check-in systems usually comprise many hardware components intended to efficiently streamline the check-in procedure. These elements are arranged thoughtfully throughout the airport terminal to guarantee passenger accessibility and comfort. The following are some of the main environmental features of the hardware used in airport self-check-in systems:

Self-Service Kiosks: Passengers use these kiosks as their main method of checking in for flights. Typically, they have printers, barcode scanners, and touchscreens so that travellers may input flight details, choose their seats, print boarding passes, and tag luggage as needed.

Barcode/RFID Scanners: These scanners are used to read boarding pass information contained in barcodes or RFID tags. They are integrated into the self-service kiosks. This enables smooth passenger authentication and verification.

**ii. Peripherals (Input / Output interfaces if any required)**

In airport self-check-in systems, various peripherals serve as input and output devices to facilitate the check-in process efficiently. These devices interact with passengers to collect necessary information and provide relevant feedback. Here are some examples of input and output devices commonly used in airport self-check-in systems:

**Input Devices:**

1. Touchscreen Displays
2. Barcode/RFID Scanners
3. Biometric Scanners
4. Keyboards or Keypads
5. Card Readers

**Output Devices:**

1. Printers
2. Visual Displays
3. Speakers or Audio Outputs
4. Status Indicators

By incorporating a combination of input and output devices, airport self-check-in systems create a user-friendly and interactive experience for passengers while streamlining the check-in process and enhancing operational efficiency.

**People**

Several key stakeholders are involved in the development, implementation, and operation of airport self-check-in systems. These stakeholders collaborate to ensure the successful deployment and functionality of the systems. Some of the main people involved include:

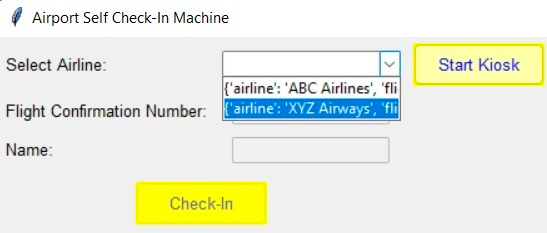
1. Airport Authorities
2. Airlines
3. Technology Providers
4. Software Engineers
5. Maintenance and Support Staff
6. Passenger

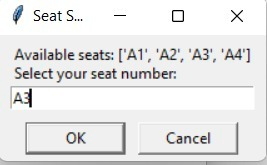
**Interfaces**   
Airport self-check-in systems typically utilize a combination of hardware and software interfaces to facilitate the check-in process efficiently.

1. **Interface with Database**

The interface with the database is crucial for airport self-check-in systems as it facilitates the storage, retrieval, and management of passenger and flight-related information. Here's how the self-check-in system interfaces with the database:

1. Data Storage
2. Data Retrieval
3. Data Update
4. Integration with Backend Systems
5. Transaction Processing
6. Data Security
7. Scalability and Performance

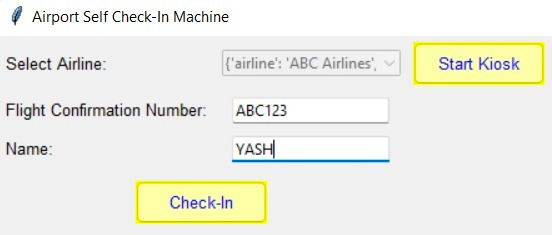


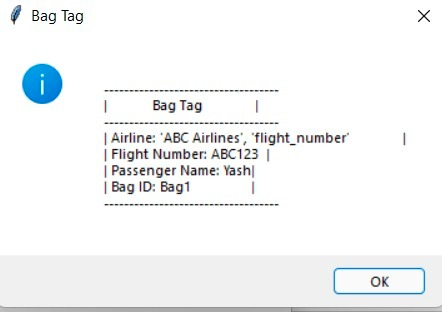


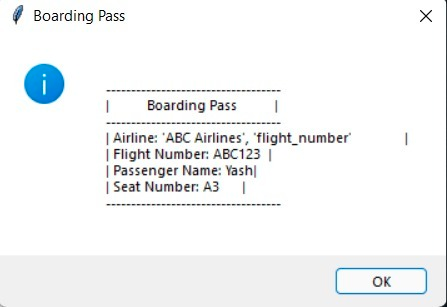
1. **Interface with User**

The interface with the user is a fundamental aspect of airport self-check-in systems, as it directly influences the user experience and ease of use. Here's how the self-check-in system interfaces with the user

1. Graphical User Interface (GUI)
2. Touchscreen Interaction
3. On-screen Instructions
4. Language Selection
5. Accessibility Features
6. Error Handling and Feedback
7. Confirmation and Feedback
8. Assistance Options







**Constraints**

Airport self-check-in systems have revolutionized the way travelers check in for their flights, offering convenience and efficiency. However, they do come with some limitations:

as per our project there are few constraints that are mentioned below :-

1. the PNR number of the passenger must be in caps locks
2. The name of the passenger should match with the name as in the e-ticket number
3. the passenger must the number of bags as per the airline policy to be checked in
4. The passenger must check all the information entered before proceeding of the printing of the baording pass and bag tags

**Functional Requirements**

The following are typical functional requirements for an airport self-check-in system:

**1.User Verification**:

Users should be able to safely authenticate themselves via the system by entering a booking reference number, scanning a passport or boarding ticket, or utilising biometric authentication (such as fingerprint or face recognition).

**2.Choosing a Flight**:

From a list of available flights, users ought to be able to choose their preferred flight.

For every flight, the system ought to show pertinent data including the destination, gate number, and departure time.

**3.Handling Baggage**:

It should be possible for users to specify how many and what kind of bags they are checking in.

For every item of checked luggage, the system ought to produce boarding tickets and baggage tags.

**4.Choosing a Seat**:

When self-checking in, if the airline permits seat selection, the system need to Give consumers access to a graphical seat map so they may select where they want to sit.

**5.Printing Documents**:

Boarding permits, luggage tags, and any other required documentation (such as a confirmation of the passenger's visa or a gate pass for a companion) should be able to be printed by the system.

Processing of payments (if relevant):

Should luggage or seat selection incur fees, the system need to enable safe payment processing.

Connectivity with Airport and Airline Systems:

To get flight and passenger information, the self-check-in system has to be integrated with the airline's reservation system.

In order to update aircraft status, gate assignments, and baggage handling information in real-time, it should also interact with airport systems.

Language and Options for Accessibility:

To serve foreign visitors, the system ought to support several languages.

Users with disabilities should be able to use it by following the applicable guidelines for accessibility.

Handling Errors and Validation:

User inputs should be verified by the system to guarantee correctness and completeness.

When there are errors in the system or when input is not legitimate, clear error messages ought to appear.

Security Procedures:

Strong security measures should be used by the system to safeguard user information and stop illegal access.

It need to adhere to pertinent data protection laws, such as HIPAA or GDPR, based on the jurisdiction.

Help and Encouragement:

For users who run into problems during the self-check-in procedure, the system ought to include help and support alternatives like help buttons or on-screen chat support.

Verification and Comments:

The self-check-in procedure should be completed successfully, and users will receive a confirmation message and any necessary instructions from the system.

It should be possible for users to offer comments on their impressions on the self-check-in mechanism.

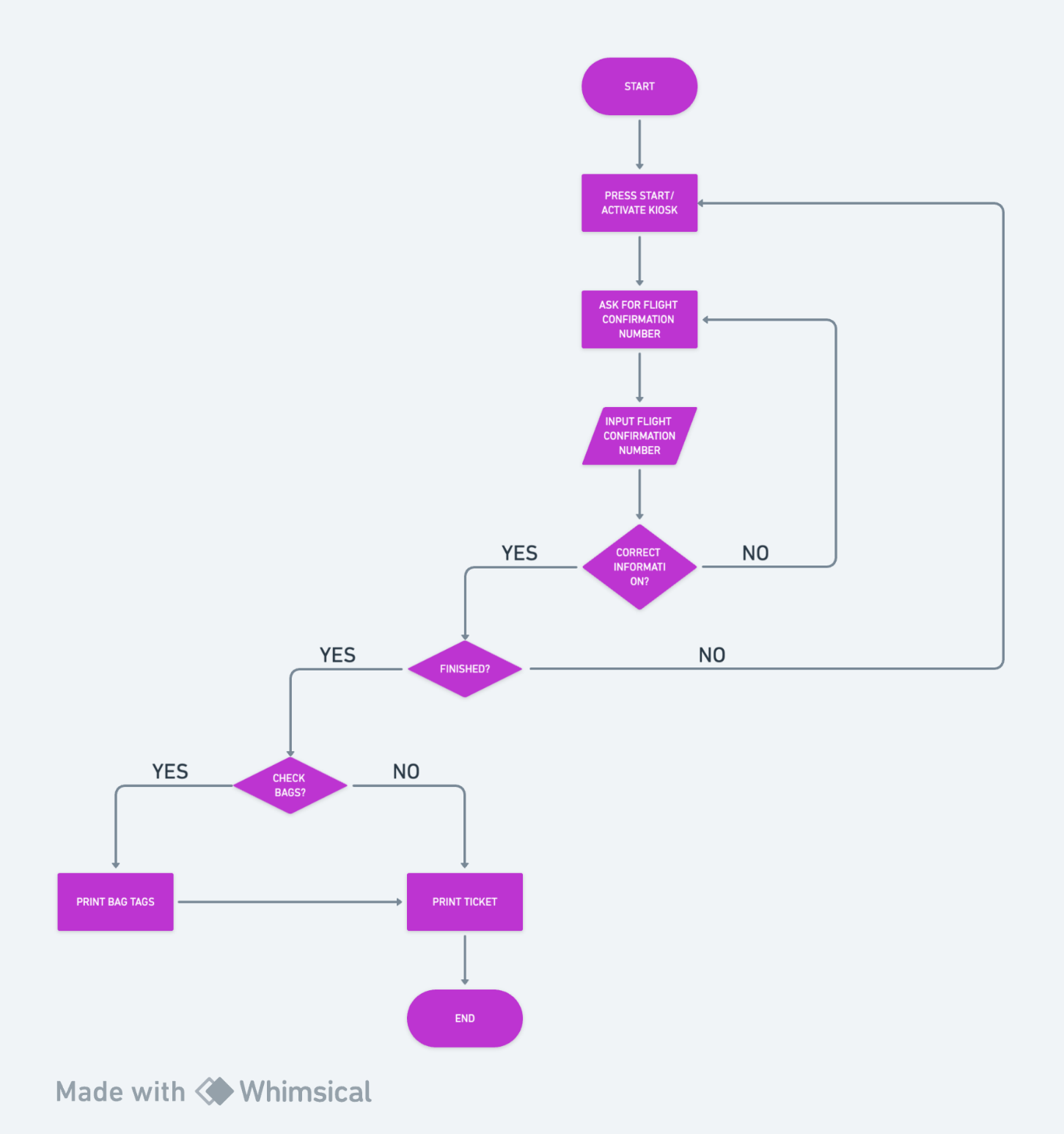
These functional specifications guarantee that the self-check-in airport system is easy to use, effective, safe, and complies with all applicable laws and industry standards. They also help passengers have a more seamless and easy time when travelling through airports.

**Non functional requirements**

Non-functional requirements for an airport self-check-in system define the qualities or attributes that the system must possess, rather than specific functions it must perform. These requirements focus on aspects such as performance, usability, reliability, security, and scalability. Here are some examples:

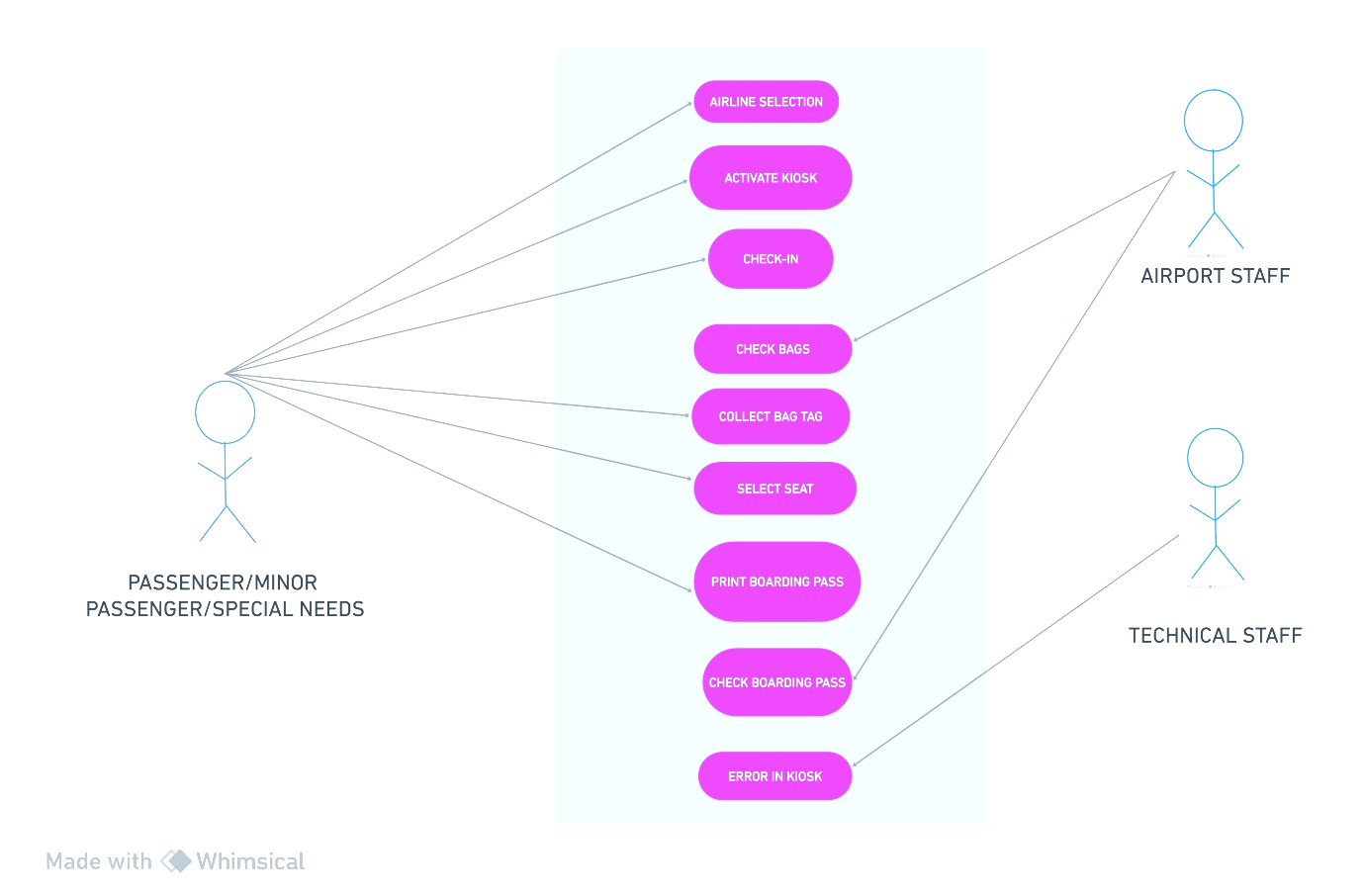
1. Performance
2. Usability
3. Reliability
4. Security
5. Scalability
6. Interoperability
7. Maintainability
8. Performance Efficiency
9. Regulatory Compliance
10. Backup and recovery

**FLOWCHART**



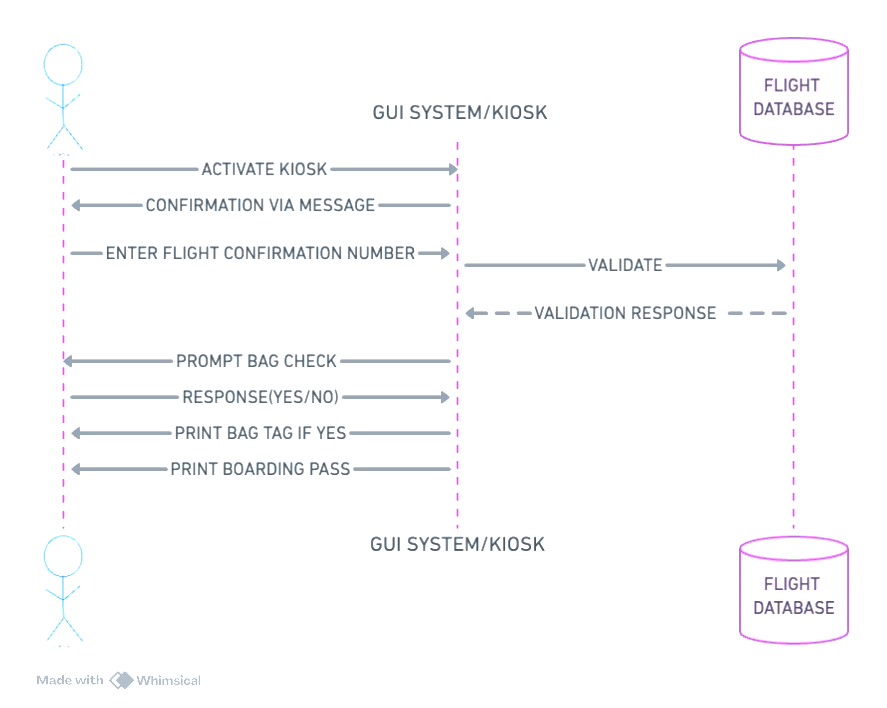
The flowchart has been made to show the self-check-in process. In this there is a start process where the passenger will need to press the start button to start the check-in process. After the passenger has successfully pressed the start button the passenger then needs to the PNR(Passenger Name Record ) number that has been assigned with their booking. If the provided PNR number is correct the passenger then will get directed to a page where they need to select the number of bags they need to check-in. if there are no bags that needs to be checked in then it will finish the process of check-in and print the boarding pass. Else the machine will generate the bags tags along with the boarding pass. The passengers then need to apply the bag tags on their bags that they wish to check-in an then they can put the bags in the convey for it to reach it to their respective flight. After this the check-in process is complete and they can proceed for the security check.

**USE-CASE DIAGRAM**



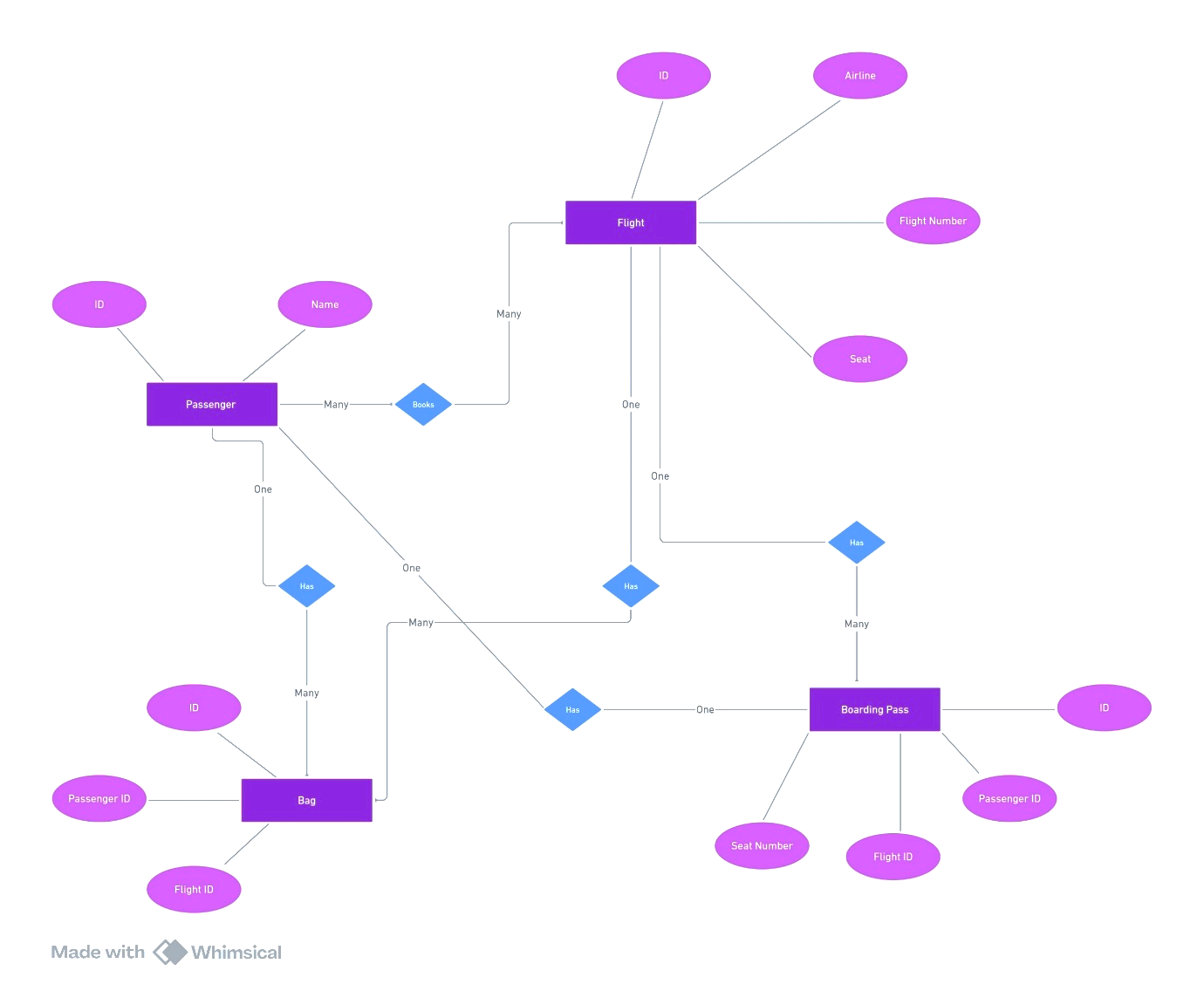
The following graphic illustrates the requirements of different stakeholders in an airport self-check-in system. The characters fulfil the conditions listed below for a self-check-in system. There are essentially three characters in the figure above: a traveller, an airport employee, and a technical staff member. The majority of the task is being completed by the traveller; only the airport personnel and technical professionals provide assistance when needed. The traveller prints the boarding pass and the bag tags for the luggage to be checked in after first choosing the airline and beginning the self-check-in procedure. Following this, the airport personnel inspects the luggage that require checking in. the technical staff member is basically there to assist the passenger if the self check-in machine fails.

**SEQUENCE DIAGRAM**



The project's sequence diagram may be seen in the figure above. The process begins at the registration kiosk when the traveller enters the necessary information. A confirmation message is then delivered to the traveller through message. The system is then supposed to submit the data to the flight database for validation after the assailant has entered the flight number. The quantity of luggage to be checked in is prompted by the kiosks following the successful confirmation from the airline database. after which the passenger must answer "yes" or "no." As a result, the kiosks print the luggage tags and provide the user with a boarding card.

**ER DIAGRAM**



The project's ER diagram is shown in the diagram above. The passenger class is where the ER diagram's flow begins, and it has several related classes inside(passenger, flight etc). The traveler has the ability to make several flight reservations at once, check in one or more baggage and print only one boarding ticket. The following characteristics are assigned to each class and associated classes:-

1. Passenger -> can have name and his identity

2. Flight: The flight number, airline name, flight ID, and seat number may be available.

3. Bags -> may contain flight\_id, passenger\_id, and bag\_id.

4. Boarding pass: This contains your seat number, passenger ID, and flight ID.

**Miscellaneous**

**We have used the following abreviations nd references in designing the succesfull project on the topic airport self check in system:**

1. **KIOSK=** it is a machine that is used by the passengers for self check in purpose and enables the passenger to generate the boarding pass and the bag tags without any interaction with the airport emplyees.
2. **BOARDING PASS=** A paper document that contains the necessary information of the passenger for smooth boarding process at the airport
3. **BAG TAGS**= a small paper document that is attached to the chek in bags of the passenger that has the flight number and the pnr of the passenger that can clearly identify which flight the bag needs to go to.

**REFERENCES:**

1. class diagram study material

URL:<https://www.geeksforgeeks.org/class-diagram-for-airport-management-system/?ref=header_search>

1. A Guide to Airport Self Check-in Kiosks

URL:<https://www.aratek.co/news/a-guide-to-airport-self-check-in-kiosks#:~:text=An%20airport%20self%2Dcheck%2Din,and%20verifying%20the%20traveler's%20ID.>

1. Application used for making diagrams

URL: <https://whimsical.com>