# 

Diploma in Software Engineering

# CT610 Software Engineering

Student Name: Yevheniia Tychynska

Student ID: 23105729

# Assignment: Final Exam

Date of Submission: 10/03/2024

Number of Pages: 49 (*Inc. Cover page, table of contents and references*)

Table of Contents

[Question 1 4](#_Toc160906035)

[Scenario: AeroRewards - Next-Gen Air Miles Program 4](#_Toc160906036)

[Introduction 9](#_Toc160906037)

[Assumptions 10](#_Toc160906038)

[Part A - Project Scope Specification Document 11](#_Toc160906039)

[1. Requirements engineering process. 11](#_Toc160906040)

[2. Use Cases 13](#_Toc160906041)

[3. Functional requirements 14](#_Toc160906042)

[4. Register for AeroRewards (Platinum tier) use case description. 17](#_Toc160906043)

[Part B – Non-functional requirements and constraints 20](#_Toc160906044)

[1. Non-functional requirements list. 20](#_Toc160906045)

[2. The most important non-functional requirements explained 21](#_Toc160906046)

[3. Project constraints 22](#_Toc160906047)

[Part C – Software testing 25](#_Toc160906048)

[1. Testing strategy 25](#_Toc160906049)

[2. Testing activities 27](#_Toc160906050)

[3. Examples of a test specification 30](#_Toc160906051)

[Part D - Risk Management Plan 36](#_Toc160906052)

[1. Project risk register 36](#_Toc160906053)

[2. Failure Modes and Effects Analysis (FMEA) 38](#_Toc160906054)

[3. Project Risk Mitigation & Contingency Plan 42](#_Toc160906055)

[4. Key Risk Information Sheets 44](#_Toc160906056)

[Conclusion 47](#_Toc160906057)

[References 48](#_Toc160906058)

# Question 1

## Scenario: AeroRewards - Next-Gen Air Miles Program

Your company has been contracted to design and implement a cutting-edge software solution for a major airline's revamped air miles program, named "AeroRewards." The airline is redefining its loyalty program to stay competitive and enhance customer engagement. Your task is to deliver a modern and secure system that caters to the expectations of the modern traveller.

**Project Overview:**

AeroRewards aims to transform the traditional air miles program into a dynamic and personalized experience for frequent flyers. The system will offer advanced features, seamless integration with travel and (corporate) credit card services, and a user-friendly interface to enhance customer satisfaction and loyalty.

After a preliminary meeting with management the following notes on some of the project requirements have been made:

**1. User Registration and Onboarding:**

* Users can register for AeroRewards through the airline's website or mobile app.
* During onboarding, users provide basic details, such as name, contact information, preferred communication channels, an associated credit card number and a corporate account number (if applicable).
* Two factor authentication using a mobile phone number applies to all new accounts with optional integration with facial recognition for streamlined registration and enhanced security.

**2. Tiered Membership Levels:**

* AeroRewards introduces tiered membership levels (Silver, Gold, Platinum) based on users' travel frequency, spending and corporate status which are specified by the user on registration.
* Both Gold and Platinum members must have an associated AeroRewards credit card which is linked to a corporate account setup separately with the airline marketing team.
* Platinum membership is only available to individuals with an annual spend of $80K or more while Gold members must have an annual spend of $50K or more. Silver members have no annual spend requirement but must commit via the registration terms and conditions to take at least 4 long haul flights a year.
* Members enjoy tier-specific benefits, such as priority boarding (All levels), lounge access (Platinum and Gold only) and exclusive longhaul discounts (Platinum only).

**3. Point Accumulation And Redemption:**

* Within the app members can view their account details, the total number of points accumulated to date and a history of each point related transaction.
* Members accrue points not only for flight bookings but also for diverse interactions with the airline, including in-flight purchases, booking accommodations, and engaging in partner programs.
* The actual transactions for flight bookings, in flight purchases and booking accommodations do not take place within the app.
* Points are automatically accumulated if payment is made for any service via the card stored in the AeroRewards app.
* AeroRewards credit card holders earn double points for every airlinerelated expenditure and 1 point for each euro spent on purchases outside the program.
* Points are redeemable within the app for flights, seat upgrades, hotel stays, and lifestyle products.

(i) When users select the "Redeem Points" option in the AeroRewards app, external APIs utilize artificial intelligence to analyze users' travel history and preferences, generating a list of personalized recommendations for point redemption.

(ii) Upon selecting the "Redeem Now!" button next to the personalized recommendation, an external API takes charge of the purchase, whether it is through the company's flight purchase system or a third-party provider, such as accommodation services.

AeroRewards is committed to stringent data protection measures to ensure privacy and compliance with international regulations including multi-factor authentication for secure account access and regular security audits and vulnerability assessments.

Please note any further assumptions or additions you feel necessary regarding the system specification as you answer the following questions:

For the above system:

**Part A**

Imagine you are the lead software engineer/project manager for this project. Using lessons learned from the CT610 module and your knowledge of the project scenario above, create a Project Scope Specification Document.

Your document should include:

(i) The requirements process that you will follow. Include details of any requirements elicitation techniques you will apply.

(ii) For your first meeting with the requirements engineer, create a Use Case Diagram based on the scenario above detailing the main behaviour of the Aerorewards system. Include all Actors and Use Cases suggested by the system description.

(iii) For your first meeting with the stakeholders, compile a draft Aerorewards requirements specification list using a prioritisation technique. Your answer should include explanation of the technique and justification of priorities where necessary.

(iv) Provide a full Use Case Description with scenarios for the “Register for Aerorewards (Platinum tier)” use case.

**Part B**

Specify the non-functional requirements and constraints you would expect in this project. Identify the three most important non-functional requirements and explain why they are most important.

**Part C**

(i) Describe in detail (and justify), the testing strategy which you would adopt for this project.

(ii) Describe the various activities which you would plan as part of implementing this strategy, when they would be performed, their inputs and outputs, and who would be involved.

(iii) Include at least 2 examples of a test specification.

**Part D**

Develop a risk management plan for the design and implementation of the AeroRewards system. In this plan be sure to consider factors such as user registration security, integration with partner systems, and the use of AI for personalized recommendations.

Your plan should include:

(i) A Project Risk Register

(ii) A Failure Modes and Effects Analysis (FMEA)

(iii) A Project Risk Mitigation & Contingency Plan

(iv) Key Risk Information Sheets

(University of Galway 2024).

# Introduction

Airline loyalty program usually worth more than an airline itself (AltexSoft 2022). Considering that this is a crucial part of the business, it is very important for a company to have useful application for a frequent flier program in modern high-tech world. Such application must be intuitive, secure, and reliable, as keeping users highly satisfied is the only way to make them loyal.

In this paper, I as a software engineer focused on these the most important characterictics while planning the AeroRewards project. The project team goal is not only create a frequent flier program application, but also ensure project success. The important success criteria are the following:

* Scope: it should be clear for the project team what they work on.
* Timeframe: the project must be completed in time.
* Budget: the project must be completed within the budget.
* Client expectations: client goals should be prioritised.

To be able to achieve project completion whithin time, budget, and scope, it is very important to plan it thoroughly: specify requirements, plan validation and verification, assess risks. These three aspects are presented in this paper which is divided into four main sections: functional requirements, non-functional requirements, test plan, and risk management plan.

# Assumptions

1. Any communications with external to AeroRewards app systems are performed using APIs. For example, anytime when credit card registered in AeroRewards is used for partner’s purchases, system gets information about a partner where transaction was performed, date/time of transaction, and spend sum. AeroRewards system knows nothing about third parties’ systems but it is connected to an API that fetches neccesary information from these systems and passes it to the AeroRewards system. The same applies to any communication with the airline’s systems and third parties’ systems.

2. Credit card must be verified during registration process to make sure that it exists and it’s not blocked.

3. AeroRewards application should comply with Web Content Accessibility Guidelines to ensure that all the airline’s customers can use the program (World Wide Web Consortium 2023).

4. If customer wishes to register for the program, has annual spend that is enough for Gold or Platinum level, but do not have a corporate account, they might be temporary registered as a Silver member. In this case, their data is sent to the airline’s marketing department for corporate account setup.

5. As the AeroRewards system deals with travelers all around the world, the application should implemet localization. This means the system at least should support multiple languages and currencies (AlexSoft 2022).

# Part A - Project Scope Specification Document

## 1. Requirements engineering process.

The requirements engineering process that I will follow for this project will be incremental. That means that three key activities which are elicitation, specification, and validation will interleave, as per Sommerville (2016).

At the first stage of the requirements engineering process, a business analyst and I will focus our efforts on high-level business requirements and their analysis. We need to understand whether all the business requirements feasible or not. Then, we’ll need to gather, analyse, and specify user requirements. After that, it is important to identify the system non-functional requirements. However, all these requirements might be changed and specified at the later stages of the system developing. There are two reasons behind it:

* All requirements change (Sommerville 2016).
* The system will be developed following agile methodology which makes incremental development of the system and the requirements possible.

Although incremental requirements engineering process adds some degree of uncertainty, it positively impacts the stakeholder’s understanding of what the system should do and how it should be done.

During requirements elicitation, I as a software engineer will meet with the stakeholders to get more information about the features they want, the hardware constraints, and the stakeholders’ work activities. (Sommerville 2016). I found it very important to gather the AeroRewards app’ users (airline’s customers) expectations as their loyalty will depend on how they will be satisfied with the application.

The stakeholders that I will meet with are the following:

* Airline management,
* Airline customers,
* Corporate clients,
* Airline marketing team,
* Airline compliance team,
* Partner organizations (e.g., credit card issuers, hotels, etc.),
* The project IT team.

I will use several requirements elicitation techniques for the AeroRewards project:

1. Interviews with the airline management, marketing team, and corporate clients to gather their expectations towards the app.
2. Meetings with IT team members to define system non-functional requirements.
3. Interviews with airline compliance team to take into account all the regulations towards the app, including financial and data protection.
4. Interview (possibly remote) with focus group (diverse range of customers) to identify their expectations, preferences, and concerns towards the AeroRewards project.
5. Observation of how the customers interact with the current airline’s website and app now to understand their issues and pain points.
6. Studying concurrent airlines’ applications with aim to find useful features or some points that might be done better.

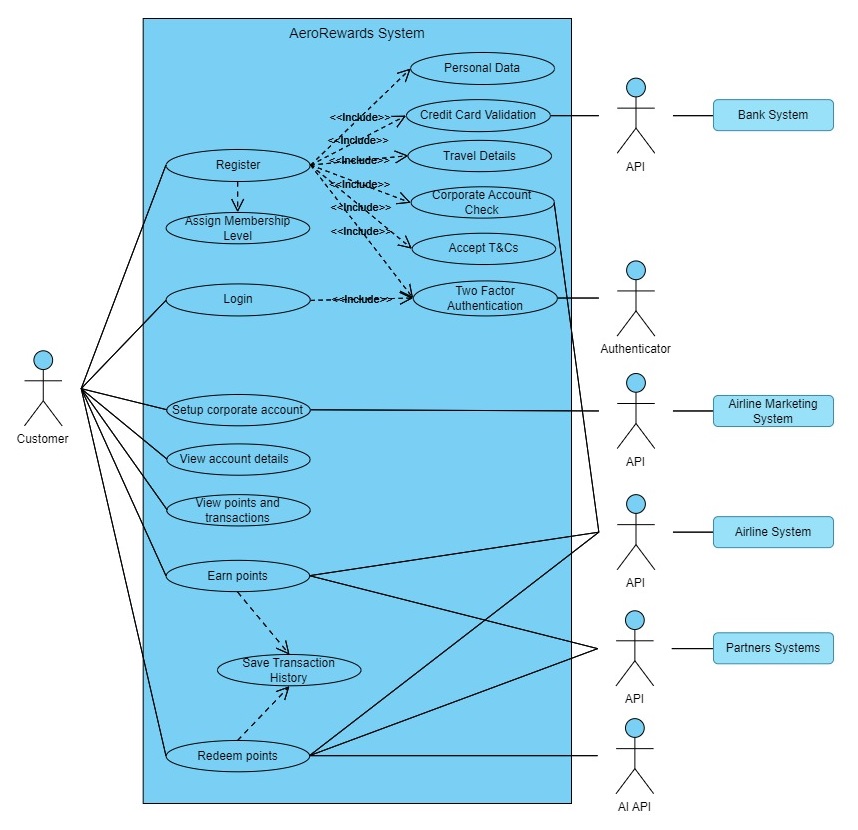
All the gathered requirements will be analysed and written using natural language in the requirement specification document for further discussion with the stakeholders. All the requirements will be validated using reviews and test cases.

## 2. Use Cases

I have identified the following actors of the system:

* Customer,
* Authenticator,
* External APIs that include AI API and other APIs for communication with third party providers (such as hotels and lifestyles products sellers) and for interactions with the airline systems (marketing department, flight booking system, in-flight purchases system and so on).

For the first meeting with a requirements engineer, I have created a use case model of the overall system that reflects the system behaviour for user registration and onboarding, tiered membership levels, and point accumulation and redemption. This diagram is presented below.



*Figure 1. AeroRewards system use case*

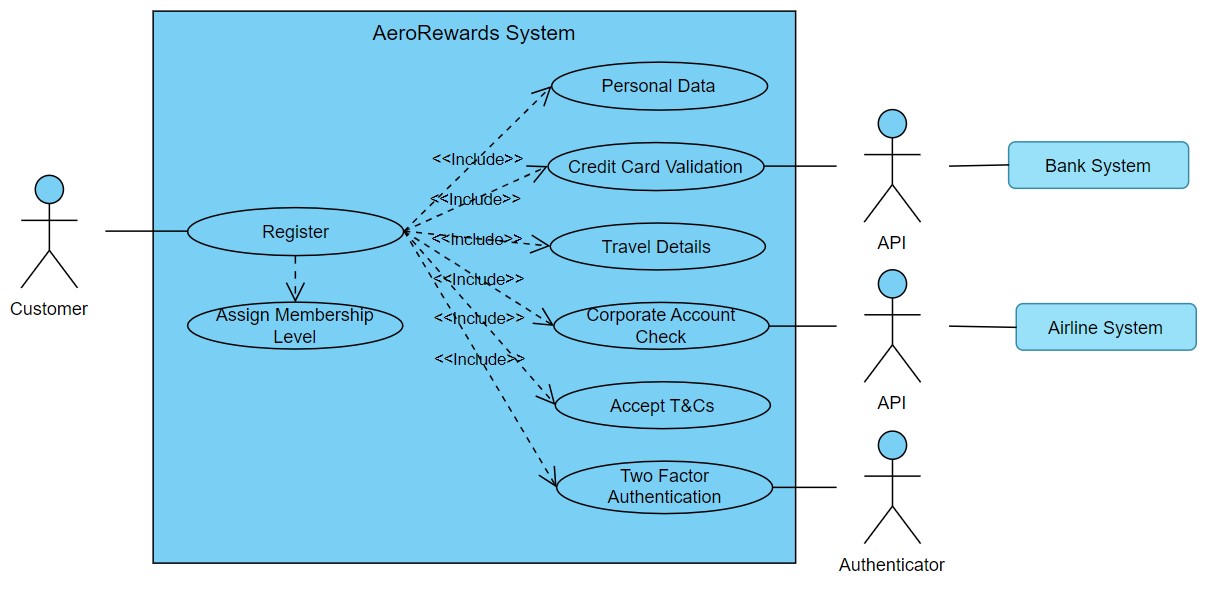
## 3. Functional requirements

After analysing high-level business needs, I have created a draft of the requirements specification for further discussion and negotiation with the stakeholders. The requirements here are sorted using MoSCoW prioritisation technique where the requirements are broken into four categories: must have (the requirements without which product will not work or will not have sense), should have (important but not vital requirements), could have (not necessary but can have small impact on the system), and will not have or would be nice to have (requirements that might be left aside for the current release without impact on the system) (Fernandes 2023).

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Classification (MoSCoW)** | **Comments** |
| FR1 | System shall allow the airline customers to register for AeroRewards through the airline's website or mobile app. | Must have | User ought to be registered to participate in AeroRewards program. All the provided information is used for the experience customization and user’s data security. |
| FR2 | Customer shall provide their basic details: name, contact information, preferred communication channels, a credit card number, and a corporate account number (if applicable) – to be able to register. | Must have |
| FR3 | Customer shall provide travel frequency, spending and corporate status during registration process. | Must have |
| FR4 | Registered user shall have to log onto AeroRewards using their username and password from registration process as well as a one-time password texted to their mobile phone. | Must have | These two requirements are about data protection, so it is non-negotiable. |
| FR5 | User shall complete two factor authentication using a mobile phone number during registration and every log on. | Must have |
| FR6 | System shall have three membership levels: Silver, Gold, and Platinum. Program’s features and benefits available/visible to user are level-based. | Must have | Airline miles program is usually tired. It helps to keep customers loyal. |
| FR7 | System shall define user’s membership level based on their travel frequency, spending and corporate status. | Must have |
| FR8 | System shall integrate with the other airline systems for flights booking, in-flight purchases, marketing department, etc. | Must have | This feature allows to control customer’s spending on flights. |
| FR9 | The system shall store all the registered customers’ account in its database. | Must have | Having separate secure database is vital. |
| FR91 | User shall accept terms and conditions to be able to register. | Must have | This is a registration policy, non-negotiable. |
| FR10 | Gold and Platinum members should have AeroRewards credit card and corporate account. | Should have | In my opinion, this requirement is important but non-vital. |
| FR11 | System should define a customer as a Platinum member if they have an annual spend >= $80K, as a Gold member when they have an annual spend >= $50K. | Should have | Memberships levels’ definition might be changed. Although, these requirements are important, these conditions may be reviewed. |
| FR12 | System should define any customer that commit via the registration terms and conditions to take at least 4 long haul flights a year as a Silver member. | Should have |
| FR13 | System should allow priority boarding to all its levels, lounge access to Gold and Platinum members, and long-haul discounts to Platinum members. | Should have | These requirements are important for customer’s satisfaction but non-vital for the system. |
| FR14 | Registered customers shall be able to view their account details, the total number of points accumulated to date, and a history of each point related transaction. | Should have |
| FR15 | System should add points to member’s account for any purchases within the airline and partner programs. | Should have |
| FR16 | System should integrate with partners’ systems to allow members earn points for booking accommodations and engaging in partner programs. | Should have |
| FR17 | System should automatically add double points for airline related purchases and 1 point for each euro spent on purchases outside the program to AeroRewards credit card holders. | Should have |
| FR18 | Users should be able to redeem points within the app for purchases related to the airline (flights, seats, in-flights purchases) and the partners programs (lifestyle products, hotels, etc.). | Should have |
| FR19 | System should integrate with external API that takes charge of the purchase under points redemption. The API should connect to the airline and partners systems. | Should have | Important requirements for transaction’s protection. |
| FR20 | System could support multiple languages and currencies. | Could have | Localisation is often required for airlines’ systems |
| FR21 | System could integrate with facial recognition for enhanced security. | Could have | Not necessary but can impact how logon will be realised. |
| FR22 | System could provide user with personalised recommendations for point redemption using AI. | Could have | Not necessary but can improve user’s satisfaction with points redemption. |
| FR23 | System could have localization feature: multiple languages, currencies, cultural features support. | Could have | Not necessary but often use in apps for travellers. |

The requirements will be discussed with the development team for further fine specification to ensure that nothing is missed based on the available client expectations and wishes. And then, the requirements will be brought up for discussion with the client.

## 4. Register for AeroRewards (Platinum tier) use case description.



*Figure 2. Register for AeroRewards use case diagram.*

|  |  |
| --- | --- |
| **Register for AeroRewards (Platinum)** | |
| **Actors** | Customer, Airline Marketing Team, Authenticator |
| **Description** | Customer registration can flow from the airline’s website or mobile app. In both situations, the registration use case starts from the main page where “Register Now” button is presented.  **Flow of events:**  1. User presses “Register” button.  2. The system displays screen with fields to be filled by user. The fields include user’s full name, phone number, email, address, credit card number, and preferred communication channel.  3. User enters personal information.  4. User is asked to enter one-time password from a received text to confirm identity using 2FA.  5. User enters a one-time password.  6. System checks whether the entered credit card exists and is not blocked.  7. System asks user to specify users' travel frequency and annual spend.  8. User chooses a range from the dropdown menus.  9. System checks user’s annual spend in the airline’s database to determine whether it is $80K and more.  10. System asks to enter corporate account number.  11. User enters corporate account number.  12. System checks whether the user’s credit card is associated with provided corporate account number.  13. After successful validation, system asks user to read and accept T&Cs.  14. User accepts terms and conditions.  15. System offers to use facial recognition as a part of authentication process.  16. User accepts or declines the offer.  17. System displays message about successful registration.  **Alternative flows:**  1) At the step 4 of the basic scenario, if user did not receive password, they are asked to check phone number and try again.  2) At the step 6, if the credit card does not exist or is blocked, user is asked use different card. Then, use case continues.  3) At the step 9, if the annual spending is less than $80K, corresponding warning message is displayed. User is offered to register as Gold or Silver member depending on annual spend. Platinum tier registration does not continue.  3) At the step 11, user may specify that they do not have corporate account number. In this case, user is offered to register as Silver member temporary and create corporate account. User can decline or accept offer. If decline, registration is terminated. If accept, user is registered as a Silver member. System sends user’s detail to marketing department where corporate account will be setup.  4) At the step 12, the System can detect that entered credit card is not associated with corporate account. In this case, error message is displayed, and user is asked to check the provided information and try again.  5) At the step 14, if user does not accept terms and conditions, user will not be registered. |
| **Data** | Customer personal information, credit card details, corporate account detail, user’s travels details |
| **Stimulus** | User commands issued by customer |
| **Response** | Confirmation on the successful registration. |

# Part B – Non-functional requirements and constraints

## 1. Non-functional requirements list.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **NFR Type** | **Description** | **Classification**  **(MoSCoW)** |
| NFR1 | Security | System must store user’s personal information, credit card details, and transaction history securely. All sensitive information shall be encrypted in accordance with Advanced Encryption Standard 256. | Must have |
| NFR2 | Security | System shall have the ability to support two-factor authentication. | Must have |
| NFR3 | Security | System shall comply with General Data Protection Regulation. | Must have |
| NFR4 | Auditability | System should log any user’s or system’s activities, so in case of any security issues a trail can be found. | Must have |
| NFR5 | Integration | System shall integrate with a range of external APIs to ensure data exchange with an issuing bank system, partners systems, and a range of the airline’s systems (flights booking, marketing department, and so on). | Must have |
| NFR6 | Reliability | System shall be highly reliable, with 98% of availability for users in any given month. | Must have |
| NFR7 | Reliability | System shall recover from errors without data loss and without integrity compromising. | Must have |
| NFR8 | Performance | System shall be available to 5,000 users at a time with less than 3 seconds response. | Must have |
| NFR9 | Performance | System shall be able to handle at least 25,000 transactions (concurrent user registrations, points accumulations, and redemptions) per hour without decreasing in performance. | Must have |
| NFR10 | Compatibility | System shall be compatible with any types of tablets, smartphones, and PCs; any types of OS and browsers. | Must have |
| NFR11 | Scalability | System should be scalable allowing growth in the number of users and data volume increase. | Should have |
| NFR12 | Maintainability | Mean time to repair (MTTR) after failure should be no more than 3 hours. | Should have |
| NFR13 | Customizability | System could allow adding new features in the future to keep the AeroRewards system up do date with changing business environment. | Could have |

## 2. The most important non-functional requirements explained

The list of non-functional requirements presented above is based generally on the client’ (the airline company) expectations. At the first meeting with management, it was highlighted several times that the system must be secure. Hence, in my opinion, the most important functional requirement toward AeroRewards system is security. The reasons behind this importance are the following:

* Users’ credit cards will be linked to the system and used within the system which means that Payment Card Industry Data Security Standard (PCI DSS) is applied to the system as to any other merchant or service provider that uses or stores cardholder’s data (IT Governance n.d.).
* System will store customer’s personal information. This information must be protected in accordance with General Data Protection Regulation.
* Any data breach may cause not only users’ financial losses (which the company have to refund if this happens as a result of the system’s insecurity) but also a decrease in customer loyalty.

The second most important non-functional requirement is the system’s integrity. The reason behind this opinion is based on the notes made at the meeting with management. Considering that credit card will be used in the system, Platinum and Gold member’s credit card will be linked to the corporate account setup by the marketing department (which is presumably happens in the airline’s another system), customer will be able to earn points for purchases from the airline’s partners (third party), for flights booking (separate system) and in-flights purchases (different system), and the same applies to the points redemption, the system must integrate with a lot of other systems. For the fastest and the most convenient solution and according to the client’s wishes, external APIs will be used.

And the third most important non-functional requirement, in my opinion, is the system reliability because:

* The client highlighted that one of their goals with the AeroRewards application is customer’s satisfaction: “enhance customer engagement”, “caters to the expectations of the modern traveller”, “to enhance customer satisfaction and loyalty”. Keeping customer satisfied impossible if even the most intuitive application is unreliable and continuously crashes. For example, customer relies on the AeroRewards app to access their loyalty points during booking process (as a discount); booking time is usually restricted to around 15 minutes; if the application often crashes, there is a high probability it will do it at this the most inconvenient time.
* Reliability supports system’s security: as the app handles customers’ sensitive data, frequent crashes increase risk of data loss or corruption.
* Unreliable system’s support and constant troubleshooting are much more expensive than creating highly reliable system at the first place.
* Frequent outages can negatively affect the airline’s brand reputation and also the IT company’s brand reputation.

## 3. Project constraints

As usually, I have divided the AeroRewards project constraints into two categories: business constraints and technical constraints. The former includes time, budget, scope, resources, market changes, and so on that can affect the project development, whereas the latter is the more specific to the project. The list of the project constraints is presented below.

|  |  |  |
| --- | --- | --- |
| **ID** | **Description** | **Category** |
| BC1 | The project is constrained by its scope which will be defined and signed by the stakeholders at the beginning of the project execution. Although changes in the requirements specifications are allowed, any significant additions may lead to the scope creep and affect project budget. | Business constraint (scope) |
| BC2 | Presumably the budget is limited to 500,000 euro which impose limitations on project progress by restricting effort that might be applied to the work and the amount of work that might be done. | Business constraint (budget) |
| BC3 | It is assumed that the project’s timeframe is 9 months. It limits the project’s scope as it might become obvious that not all amount of work can be done in this time. | Business constraint (time) |
| BC4 | Human resources are limited to one project manager, two software engineers, two QA, two BA and five software developers. If more human resources are needed, outsource should be considered. | Business constraint (human resources) |
| BC5 | Additional expertise in artificial intelligence, cybersecurity, 2FA, and so on might be required. | Business constraint (human resources) |
| BC6 | CASE tools will be used for faster project delivery. They require additional spendings. And there is a risk of integrity issues. These points must be carefully considered. | Business constraint (software resources) |
| BC7 | During the preliminary meeting with management, it was said that one of the goals of the new application is to stay competitive. This creates additional constraint as it requires some degree of innovation and market changes tracking so the product stays competitive at the time of its launch (i.e., no one would have launched application with the same features). | Business constraint (market competition) |
| BC8 | Different countries’ cultural features, laws, traditions, languages, and standards should be considered, as the system will work with travellers around the world. It creates additional layer of technical work and/or expenses and must be taken into account. | Business constraint (time, budget, resources) |
| TC1 | The AeroRewards system has to be integrated with other systems of the airline, partners, financial institutions (credit cards issuing banks), authenticator, and AI server. Even though it will be done using APIs, the system must meet certain technical specifications to use APIs which creates technical constraint for the system. | Technical constraint (external systems dependency) |
| TC2 | The system should be able to implement two-factor authentication which means that the system should be compatible with 2FA mechanism especially if it requires specific hardware and software components. | Technical constraint (2FA compatibility) |
| TC3 | Implementing 2FA and face recognition mechanisms may add complexity into the system architecture and the development process. | Technical constraint (security mechanisms integration) |
| TC4 | The system should compile with several data protection standards and law, such as GDPR, PCI DSS, and Web Content Accessibility Guidelines. | Technical constraint (security regulation) |
| TC5 | System shall be compatible with any types of tablets, smartphones, and PCs, any types of OS and browsers. This creates constraints for the development process (what techniques and methods to use to ensure that application will work on any hardware and software). | Technical constraint (application compatibility) |

# Part C – Software testing

## 1. Testing strategy

Software testing is the one of the most important parts of a software development process as it allows to make sure that the system meets functional and non-functional requirements, does not contain bugs, errors, and defects, seamlessly integrates with other systems, and so on. Testing strategy development is the first stage of software testing which determines its further course. It allows to communicate a way the software will be validated and verified to a development team so the team would be aware of the chosen testing approach and stay consistent (Hamilton 2024).

I have identified the following objectives of the AeroRewards system testing:

* Defects identification,
* Compliance with the requirements,
* Delivering high quality product.

Thus, the software testing will cover validation of the system functionality outlined in the requirements specification section, defect testing, and non-functional testing. Both static and dynamic tests will be performed to guarantee the software quality and reliability. I defined the following testing process that will be performed:

1. Functionality testing. A set of test cases will be developed and run for all the functional requirements to ensure the software works as expected according to the customer’s expectations.
2. Security testing. As security was highlighted as the one of the most important features of the application, the set of security tests will be run to ensure the system is invulnerable and does not have flows and weaknesses. I am planning to include vulnerability scanning, security scanning, penetration testing, risk assessment, and ethical hacking (Hamilton 2023).
3. Reliability and performance testing. Stress testing, endurance testing, recovery testing, load testing, volume testing, and spike testing will be performed to ensure that the AeroRewards system does not crush under high workload, does not leak or decreases in performance over time, can recover from failures and return to normal operation, can handle large number of users and large amount of data, can survive sudden spike in workload (GeeksforGeeks 2023). These different circumstances will be artificially created.
4. Usability testing will be performed after the system will pass the QA. A focus group of the airline’s customers will be asked to use the application and provide feedback. In this way, the project team can get the most unbiased opinion about the software intuitiveness and usability.
5. Compatibility testing will verify that the AeroRewards software works as expected with different software which includes range of operating systems (Windows, Linux, macOS, iOS, and Android), browsers (Google Chrome, Microsoft Edge, Internet Explorer, Mozilla Firefox, Opera, and Safari). This verification allows to make sure that users of all platforms will get equally great experience.
6. Regression testing is also important part of the AeroRewards application testing as it helps to quickly detect and fix any problems with the system that might appear after changes and updates have been made.

The AeroRewards software testing will be carried out simultaneously with its development: once the units created and tested by developers, they might be integrated and tested by the QA team, and the development team can switch to the next set of units that should be developed yet. In my opinion, it allows to accelerate the development processes.

The described above testing strategy will allow identify and fix defects at the earliest stages of the software development, accelerate processes, and produce high-quality software.

## 2. Testing activities

The AeroRewards system testing will include the following testing activities:

1. Static testing. At the early stages of the system development, system design (high-level and low-level), use cases, and test cases will be reviewed by the project team members to verify its compliance with the software requirements. During source code development process, the code will be regularly reviewed and analysed before unit testing. Although static testing is not a crucial part of the software testing, it is beneficial for detecting issues at early stages and allows to reduce testing time and cost (Devi 2023).
2. Dynamic testing will consist of four stages:
   1. Unit testing will be performed by the developers during the system development stage. It will allow fast and clearly identify fatal bugs and errors at the early stage of the development process while each unit (module, function, section, etc.) is independent. This process will be automated by using JUnit (testing framework for Java) and HTMLUnit (useful for testing web pages) tools.
   2. Integration testing will be done by QA team. At this stage created and tested units (that already work well separately) will be integrated and tested as a whole. The goal is to detect conflicts between units and integration-induced defects. Integration testing will be performed incrementally using bottom-up approach: testing will gradually move from low-level to high-level components. Katalon will be used for testing automation as it is useful for web, API, desktop, and mobile applications, supports cross-browser and cross-platform testing (Katalon n.d.).
   3. System testing will be performed by QA and development teams. At this stage, the system will be tested as a whole to ensure that it meets both functional and non-functional requirements, handles errors, and communicates with external APIs. Testigma tool will be used for the system testing automation (Testigma n.d.).
   4. Acceptance testing which will include regulatory acceptance, business acceptance, and user acceptance testing will be performed by regulatory affairs specialist, QA team, the client, and the airline’s customers. It is needed to guarantee that the software meets the client’s expectations, is intuitive and useful. At this stage black-box technique will be used: users will verify the application’s functionality without knowing internal structures.

The table of planned activities is presented below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type of testing** | **Testing activity** | **When?** | **Input** | **Output** | **Participants** |
| Static | High-level design review | Early design stage | High-level design, Design specification document, Functional and non-functional requirements | Review document which includes comments and recommendations | Software engineer, software developers, BA |
| Static | Low-level design review | After high-level design is approved | Low-level design, Design specification document, Functional and non-functional requirements | Review document which includes comments and recommendations | Software engineer, software developers, BA |
| Static | Use cases review | Requirements specification stage | Use case document, Functional requirements | Review document which includes comments and recommendations | BA, software engineer, the client |
| Static | Test cases review | After requirements specification, during sprint planning | Test plan, test cases, traceability matrix, Use cases, Requirements specification document | Review document which includes comments and recommendations | QA, BA, development team |
| Static | Code review | Development stage | Source code | Defect report that includes defects’ description and recommendations for corrections | Software engineer, developers |
| Dynamic | Unit testing | Development stage | Unit, test case, test data | Code coverage report, test logs, test report, defect report | Software developers, QA |
| Dynamic | Integration testing | After unit testing | Integration plan, test cases, test data | Defects report, test logs, test report | QA |
| Dynamic | System testing | After integration testing | System requirements, test cases, test plan, test data | Defects report, test logs, test report | QA, development team |
| Dynamic | Acceptance testing | After system testing, before deployment | Acceptance testing plan, test scenarios, acceptance criteria | Test report, stakeholders’ feedback | Regulatory affairs specialist, QA team, the client, the airline’s customers |

## 3. Examples of a test specification

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Report** | | | |
| Register for AeroRewards (Platinum tier) Test Case | | | |
| **Test Stage:** |  | | |
| **Test Date:** | mm/dd/yy | **System Date (if applicable):** | mm/dd/yy |
| **Tester:** | Paddy O’Connor | **Test Case Number:** | TC-RPT-003 |
| **Test Case Description:** | This is an integration test. Test verifies whether a customer can successfully register for AeroRewards (Platinum tier). Successful registration means that registration confirmation message is displayed to user, user’s details are stored in the AeroRewards database, all the benefits for Platinum members are activated for the user. | | |
| **Results:** |  |  |  |
| **Introduction** | | | |
| **Requirements to be tested:** | FR1 - System shall allow the airline customers to register for AeroRewards through the airline's website or mobile app.  FR2 - Customer shall provide their basic details (name, contact information, preferred communication channels, a credit card number, and a corporate account number (if applicable) to be able to register.  FR3 - Customer shall provide travel frequency, spending and corporate status during registration process.  FR5 - User shall complete two factor authentication using a mobile phone number during registration and every log on.  FR6 - System shall have three membership levels: Silver, Gold, and Platinum. Program’s features and benefits available/visible to user are level-based.  FR7 - System shall define user’s membership level based on their travel frequency, spending and corporate status.  FR8 - System shall integrate with (…) marketing department, etc.  FR9 - The system shall store all the registered customers’ account in its database.  FR10 - Gold and Platinum members should have AeroRewards credit card and corporate account.  FR11 - System should define a customer as a Platinum member if they have an annual spend >= $80K  FR 91 - User shall accept terms and conditions to be able to register. | | |
| **Roles and Responsibilities:** | 1. Project Manager: communication with stakeholders and managing expectations regarding testing outcomes. 2. QA engineer: test case development, test execution, progress, and results reporting. 3. Software developer: development of the software being tested; fixing identified problems; providing technical skills during test. 4. System administrator: hardware and network optimization. | | |
| **Set Up Procedures:** | Preconditions:   * The AeroRewards registration system is accessible. * The user meets Platinum tier criteria.   Test data:   * Test user details, * Test credit card details, * Test corporate account number.   Test steps:   1. User presses “Register” button in the AeroRewards web application. 2. User fills in required information. The fields include user’s full name, phone number, email, address, credit card number, and preferred communication channel. 3. User presses “Continue” button. 4. User is asked to enter one-time password from a received text to confirm identity using 2FA. 5. User enters a one-time password. 6. User chooses a range of travel frequency and annual spending from the dropdown menus. 7. User enters corporate account number. 8. User checks checkbox to accept terms and conditions. 9. User presses “Submit” button. 10. User accepts or decline facial recognition as a part of authentication process. 11. Confirmation message about successful registration is displayed.   After that user’s details are stored in the system database. | | |
| **Stop Procedures:** | 1. To cancel registration before submission page, click “Cancel” button. 2. To terminate registration, close web application. | | |
| **Environmental Needs** | | | |
| **Hardware and Software:** | * Intel Core i7-107000K 10th Gen, * 64-bit operating system such as Windows 7 and above, * RAM 16 GB, * 1024×768 screen resolution, * Java 1.4 and above, * reliable internet connection, preferably wired, with at least 10 Mbps speed. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Report** | | | |
| Points Redemption Test Case | | | |
| **Test Stage:** |  | | |
| **Test Date:** | mm/dd/yy | **System Date (if applicable):** | mm/dd/yy |
| **Tester:** | Aine MCGee | **Test Case Number:** | TC-PR-01 |
| **Test Case Description:** | This is an integration test. Test verifies whether a customer can get suggestions about points redemption and successfully redeem points. Successful test results are the following:   * User gets recommendations from AI about points redemptions that match user’s previous activities, * User uses points, * Correct number of points are withdrawn from user’s account. | | |
| **Results:** |  |  |  |
| **Introduction** | | | |
| **Requirements to be tested:** | FR14 - Registered customers shall be able to view their account details, the total number of points accumulated to date and a history of each point related transaction.  FR18 - Users should be able to redeem points within the app for purchases related to the airline (flights, seats, in-flights purchases) and the partners programs (lifestyle products, hotels, etc.).  FR19 - System should integrate with external API that takes charge of the purchase under points redemption. The API should connect to the airline and its partners’ systems.  FR22 - System could provide user with personalised recommendations for point redemption using AI. | | |
| **Roles and Responsibilities:** | 1. Project Manager: communication with stakeholders and managing expectations regarding testing outcomes. 2. QA engineer: test case development, test execution, progress, and results reporting. 3. Software developer: development of the software being tested; fixing identified problems; providing technical skills during test. 4. System administrator: hardware and network optimization. | | |
| **Set Up Procedures:** | Preconditions:   * The AeroRewards registration system is accessible. * The user is logged in to the application.   Test data:   * Test user account with accumulated points, * Set of options for points redemption.   Test steps:   1. User presses “View points” button in the AeroRewards web application. 2. The number of points is displayed. 3. User presses “Redeem points” button. 4. The list of available options for spending points is displayed. 5. User chooses an option by clicking “Redeem now” button beside the option. 6. Details of chosen option and number of points that will be withdrawn are displayed. 7. User presses “Confirm” button. 8. Message about successful transaction is displayed.   After that the number of available points should be decreased exactly on the number of points that was used. | | |
| **Stop Procedures:** | 1. To cancel redemption before transaction is confirmed, click “Cancel” button (available at any stage). | | |
| **Environmental Needs** | | | |
| **Hardware and Software:** | * Intel Core i7-107000K 10th Gen, * 64-bit operating system such as Windows 7 and above, * RAM 16 GB, * 1024×768 screen resolution, * Java 1.4 and above, * reliable internet connection, preferably wired, with at least 10 Mbps speed. | | |

# Part D - Risk Management Plan

Risk management is another crucial part of project management. Considering that one unforeseen event can destroy the entire project, threaten product, and affect business (Sommerville 2016), it is particularly important to predict as many as possible of such events. Risk management includes risks anticipation, but it is not one-time action, it is systematic process of continuous risks monitoring, analysis, mitigation. It is continuous decision-making process with aim of reducing likelihood that any event will lead to project, product, or business failure.

Risk management process involves risk identification, analysis, planning, and monitoring. Together they form risk management plan. The results of the risk identification and analysis is recorded in a risk register (Sommerville 2016). In the part D of the exam question, I am asked to develop a risk management plan for the design and implementation of the AeroRewards system. So, here I tried to focus particularly on the risks related to the software development processes (design, coding, integration, testing, etc.). As a first part of the risk management plan, I presented the project risk register. I would like to highlight that the list is not exhausted and probably will be changed and extended during project execution.

## 1. Project risk register

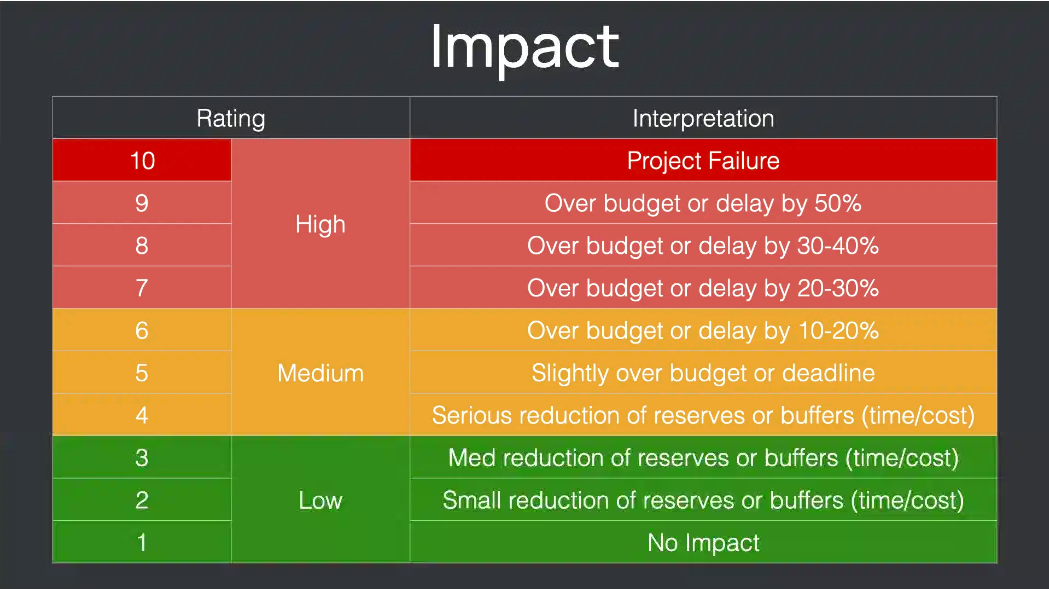
The presented project risk register includes identified risks, their impact on the project, probability of their appearance, their seriousness, and possible ways of their mitigation. The risks and their impact were identified during the meeting with the project team where all the members had a chance to use their experience in other projects to spot what can go wrong and point out the team’s weaknesses in the development of this kind of product. After that risks were analysed based on everyone’s experience. And additionally, I analysed risk management plans of the company’s previous projects, paying particular attention to those with wide usage of APIs and MFA. In this way I defined the risks probability and seriousness.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Risk** | **Impact** | **Probability** | **Seriousness** | **Mitigation notes** | **Owner** |
| 01 | Non-compatibility with authentication system for 2FA. | Delayed development, cost overrun | Medium | Serious | Specify and follow authentication system requirements before developing | Dave |
| 02 | Lack of expertise in 2FA implementation. | Delayed development, cost overrun | Medium | Serious | Training for developers asap | Dave |
| 03 | Complex UI design due to 2FA implementation | Decrease in usability | Low | Tolerable | Consider hiring a consultant | Dave |
| 04 | Incompatibility with external APIs | Delayed development, cost overrun | Low | Tolerable | Gather specified requirements for APIs before developing | Inez |
| 05 | Unexpected changes in APIs functionality | Increased development time | Medium | Insignificant | Track APIs updates | Inez |
| 06 | Increased data vulnerability due to API usage (APIs are often targets for attacks) | Additional security solutions required | High | Serious | Consider hiring highly qualified cybersecurity specialist | Alex |
| 07 | Issues with integration with partners’ systems | Mismatched data, systems incompatibility | Low | Tolerable | Meeting with partners’ IT support to clear their systems features | Mike |
| 08 | Decrease in performance due to API usage | Decrease in customer satisfaction | Low | Tolerable | Training on effective API integration for developers. System performance testing | Mike |
| 09 | Misinterpretation of membership eligibility logic due its complexity | Inconsistencies in membership levels and tiered assignments | Medium | Serious | Requirements specifications must be fine-grained | Kim |
| 10 | AI access to sensitive data | Security threat to the system | High | Serious | Implement role-based access control | Alex |
| 11 | Biased AI algorithms | Decrease in user’s satisfaction, ethical problems, the airline reputation damage | Medium | Serious | Consider hiring consultant for AI usage | Inez |
| 12 | Lack of expertise in cybersecurity | Decrease in system security | Medium | Serious | Consider hiring highly qualified cybersecurity specialist | Dave |
| 13 | Regulatory non-compliance | System will not be allowed to the market | Low | Serious | Gather and follow regulator requirements. Get legal consultant | Thomas |
| 14 | Third-party systems are not fully duplicated in the testing environment | Integration issues during deployment | Low | Serious | Collaborate with third-party IT team to get testing environment that copies their systems’ behaviour | Thomas |
| 15 | Different dependencies and configurations in production environment as opposite to testing environment | System failures and unexpected behaviour in production | Low | Serious | Copy production environment as close as possible. Practice continuous integration and deployment to catch issues as early as possible | Kim |

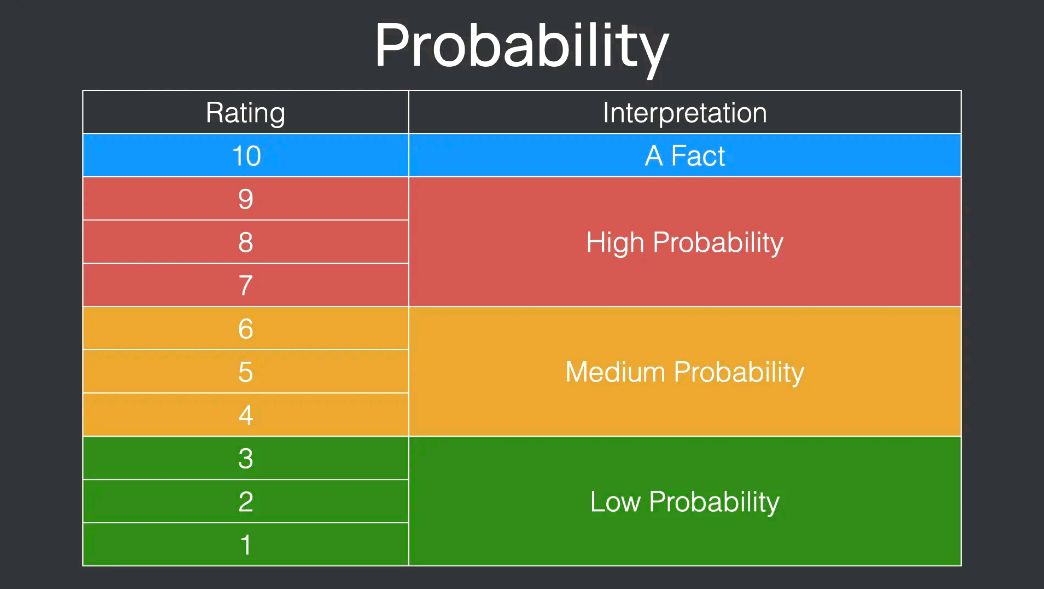
## 2. Failure Modes and Effects Analysis (FMEA)

Failure modes and effects analysis is a method of identifying potential failures, their causes, effect on a project, product, and business. Moreover, it allows to consider actions that would allow to reduce or eliminate failure (American Society for Quality n.d.). FMEA contains list of risks, their corresponding impacts, severity, probability of occurrence, and probability of detection. The latter three indicators are expressed in numbers from 1 to 10 and together determine risk priority number (RPN) and help to indicate which of the risks requires greatest attention. FMEA reflect two RPN: before risks mitigations actions and after them. This allows the actions effectiveness tracking.

According to Sommerville (2016), risk impact may be catastrophic, serious, tolerable, or insignificant. The value of risk severity usually does not change after actions taken. And risk probability can be insignificant, low, moderate, high, or very high (Sommerville 2016). This value may change after action taken. I used available in the Internet tables to express these characteristics numerically.



*Figure 3. Seriousness (impact) of risk (Nizhebetskyi 2023)*



*Figure 4. Risk probability (Nizhebetskyi 2023)*

Probability of detection value has opposite principle of assigning: failures with high probability of detection have lower values than those with low probability of detection (Hartwell 2022). Usually, this value lowers when risk mitigation actions are taken.

Risk priority number is a result of multiplication of risk’s severity, probability of occurrence, and probability of detection.

Below I present FMEA for the AeroRewards project.

*Figure 5. FMEA (see next page)*



## 3. Project Risk Mitigation & Contingency Plan

As it was already written above, knowing possible risks means that some actions might be taken to reduce the probability of their occurrence or minimize their impact on a project, product, and business. This process is known as risk mitigation. However, there a lot of risks that cannot be avoided. Then, a project team should be prepared to the worst by creating plan B. The latter strategy is known as contingency plan (Sommerville 2016). I use these three strategies (avoidance, minimizing, and contingency) carefully considering each risk which was noted in the risk register above.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Risk** | **Mitigation** | **Contingency** |
| 01 | Non-compatibility with authentication system for 2FA. | 1. Explore all available authentication systems.  2. Specify and follow authentication system requirements before developing.  3. Make sure that chosen authentication system compatible with AeroRewards technical requirements and architecture.  4. Use common protocols (e.g., OAuth).  5. Communicate with authenticator providers for support.  6. Constantly test integration between AeroRewards and authentication systems. | 1. Develop adapters to eliminate compatibility issues.  2. Hire third-party expert in MFA. |
| 02 | Lack of expertise in 2FA implementation. | 1. Organise training for developers.  2. Organise collaboration for finding best solutions.  3. Document all implementation steps and processes. | 1. Hire external consultant for leading in 2FA implementation. |
| 03 | Complex UI design due to 2FA implementation | 1. Keep usability and accessibility in mind, simplify.  2. Constantly test system intuitiveness and usability. | 1. Create interactive tutorial for users on how to set up and use the AeroRewards app.  2. Provide 24/7 online support (may be chat bot). |
| 04 | Incompatibility with external APIs | 1. Specify technical requirements of external APIs.  2. Choose APIs with flexible integration options.  3. Duplicate APIs features in testing environment. | 1. Communicate with tech support of APIs providers. |
| 05 | Unexpected changes in APIs functionality | 1. Choose APIs providers that announce any changes in versions.  2. Track APIs updates (providers announcements).  3. Perform regular regression testing.  4. Use different APIs for different functions of the AeroRewards system. | 1. Communicate with APIs providers to get support and clarifications. |
| 06 | Increased data vulnerability due to API usage (APIs are often targets for attacks) | 1. Use encryption.  2. Hire highly qualified cybersecurity specialist.  3. Use role-based access.  4. Perform security testing.  5. Choose API providers with highest security standards. | 1. Communicate problem to the stakeholders.  2. Consider additional security methods e.g., use AI for enhanced security. |
| 07 | Issues with integration with partners’ systems | 1. Meet with partners’ IT support to clear their systems features.  2. Thoroughly specify tech requirements for integration with partners’ systems.  3. Perform integration testing. | 1. Get partners’ tech support to resolve integration issues. |
| 08 | Decrease in performance due to API usage | 1. Organise training on effective API integration for developers.  2. Perform system performance testing.  3. Use cache for frequently requested data. | 1. Identify root causes of performance decrease by running additional tests.  2. Resolve root problem. |
| 09 | Misinterpretation of membership eligibility logic due to its complexity | 1. Clearly specify and document membership eligibility criteria in the requirements documents.  2. Actively communicate with the stakeholders to refine the logic.  3. Create separate use test and case test for eligibility logic.  4. Create prototype.  5. Communicate membership eligibility logic to developers. | 1. Ensure the airline representative presence on-site for consultation and fast feedback. |
| 10 | AI access to sensitive data | 1. Encrypt sensitive data.  2. Implement role-based access control.  3. Perform security audits. | 1. Cut AI access and investigate problem to find root cause. |
| 11 | Biased AI algorithms | 1. Carefully choose AI API provider, analyse how AI is trained, how it makes decisions.  2. Create special tests to identify bias.  3. Consider hiring consultant for AI usage. | 1. Change AI API. |
| 12 | Lack of expertise in cybersecurity | 1. Provide training in cybersecurity for developers.  2. Follow cybersecurity standards. | 1. Hire highly qualified cybersecurity specialist. |
| 13 | Regulatory non-compliance | 1. Thoroughly document regulatory requirements.  2. Provide regular audit.  3. Engage legal consultant and regulatory advisor. | 1. Communicate problem to the client.  2. Invest in fixing problems. |
| 14 | Third-party systems are not fully duplicated in the testing environment | 1. Collaborate with third-party IT team to get testing environment that copies their systems’ behaviour.  2. Focus on testing high-risk modules. | 1. Communicate problem to the client.  2. Find missed parts and dependencies, fix gaps. |
| 15 | Different dependencies and configurations in production environment as opposite to testing environment | 1. Copy production environment as much as possible.  2. Practice continuous integration and deployment to catch issues as early as possible.  3. Use CASE tools to make sure that testing and production environments are similar. | 1. Communicate problem to the client.  2. Find differences between testing and production environments.  3. Rework to fix the problem. |

## 4. Key Risk Information Sheets

In my opinion, API and AI present the biggest risk to the AeroRewards system when implemented and used incorrectly. Mistakes in API and AI implementation lead to security issues. And security is considered one of the most important non-technical requirements towards the AeroRewards system. Another issue with the software that I see is complex membership eligibility logic. Also, the experience of the project team shows that a situation where thoroughly tested and perfectly working in testing environment software does not work in production environment is not uncommon. Thus, I decided to present the following three risk information sheets:

* Increased data vulnerability due to API usage.
* Misinterpretation of membership eligibility logic due to its complexity.
* Different dependencies and configurations in production environment as opposite to testing environment.



*Figure 6. RIS 1*



*Figure 7. RIS 2*



*Figure 8. RIS 3*

# Conclusion

Thorough project planning with careful consideration of functional and non-functional requirements, validation and verification techniques, and possible risks is crucial for any project success. In this paper I as a software engineer have outlined all these aspects for the AeroRewards project. I’ve created a draft of a requirements specification document for further discussion with the stakeholders, listed non-functional requirements and project constraints, modelled the AeroRewards system use case diagram for discussion with the development team, described the project test strategy and test activities, specified two test cases, and developed a risk management plan for the design and implementation of the software. In my opinion, these activities and documents will be important for beginning successful project execution.

# References

AltexSoft (2022) ‘*Airline Loyalty Programs: Main Aspects, Types, and a Technological Perspective*’, available: <https://www.altexsoft.com/blog/airline-loyalty-program/> [accessed 07 Mar 2024]

AltexSoft (2022) ‘Website Localization in Travel: Key Aspects and Best Practices’, available: <https://www.altexsoft.com/blog/localization-travel-website/> [accessed 06 Mar 2024]

AltexSoft (2022) ‘*What is an API: Definition, Types, Specifications, Documentation*’, available: <https://www.altexsoft.com/blog/what-is-api-definition-types-specifications-documentation/> [accessed 07 Mar 2024]

American Society for Quality (n.d.) ‘*Failure mode effects analysis (FMEA)*’, available: <https://asq.org/quality-resources/fmea> [accessed 09 Mar 2024]

Devi, K. (2023) ‘*What is Static Testing?*’, Browser Stack, available: <https://www.browserstack.com/guide/static-testing> [accessed 03 Mar 2024]

Fernandes, T. (2023) ‘*MoSCoW Prioritization: A Guide to Managing Requirements*’, Medium, available: <https://medium.com/pm101/moscow-prioritization-a-guide-to-managing-requirements-413c5adef1cb> [accessed 05 Mar 2024]

GeeksforGeeks (2023) ‘*Reliability Testing – Software Testing’*, available: <https://www.geeksforgeeks.org/software-testing-reliability-testing/> [accessed 07 Mar 2024]

Hamilton, T. (2023) ‘*What is Security Testing? Example’*, Guru99, available: <https://www.guru99.com/what-is-security-testing.html> [accessed 07 Mar 2024]

Hamilton, T. (2024) ‘*Test Strategy Document in Software Testing (Sample Template)’*, Guru99, available: <https://www.guru99.com/how-to-create-test-strategy-document.html> [accessed 07 Mar 2024]

Hartwell, J. (2022) ‘*FMEA RPN – Risk Priority Number. How to Calculate and Evaluate?’*, iQA system, available: <https://www.iqasystem.com/news/risk-priority-number/> [accessed 09 Mar 2024]

IT Governance (n.d.) ‘*PCI DSS | What It Is and How to Comply’*, available: <https://www.itgovernance.co.uk/pci_dss> [accessed 06 Mar 2024]

Katalon (n.d.) ‘*Top 15 Automation Testing Tools’*, available: <https://katalon.com/resources-center/blog/automation-testing-tools> [accessed 07 Mar 2024]

Microsoft Security (n.d.) ‘*What is two-factor authentication?*’, Microsoft, available: <https://www.microsoft.com/en-us/security/business/security-101/what-is-two-factor-authentication-2fa#faqs> [accessed 07 Mar 2024]

Nizhebetskyi, D. (2023) ‘*Risk Management Plan Example from Real Project (+Template)*’, IT PM School, available: <https://itpmschool.com/project-risk-management-plan/> [accessed 09 Mar 2024]

Sommerville, I. (2016) *Software Engineering*, 10th ed. Harlow: Pearson Education Limited.

Testigma (n.d.) ‘*System Testing | What is it, Why, Types & How to Perform?*’, available: <https://testsigma.com/guides/system-testing/> [accessed 07 Mar 2024]

University of Galway (2024) ‘*Week 8 - CT610 Final Exam’*, available: <https://universityofgalway.instructure.com/courses/11586/discussion_topics/94866> [accessed 05 Mar 2024]

World Wide Web Consortium (2023) ‘*WCAG 2 Overview*’, available: <https://www.w3.org/WAI/standards-guidelines/wcag/> [accessed 06 Mar 2024]