BBM 384 – SOFTWARE ENGINEERING LABORATORY

SOCIAL INTEREST E-CLUB
RİSK MANAGEMENT REPORT

COPY-PASTERS

HAZAL UZAR	**21727834**
HACI KERİM ARSLAN	**21726928**
İBRAHİM KOZ	**21786303**
BAŞAK ŞÜKRAN MELAHAT ÇONTU	**21827282**
FATMA NUR DEMİRBAŞ	**21727116**

Social Interest e-Club
Risk Management Report

1 Introduction

Risk management report aims to identify, evaluate, analyze, monitor and mitigate the risks that may occur in the system. This report consists of risks and the ways to mitigate them. Mitigating the risks are important for the flow of the project; well-managed risk significantly increase the likelihood of project success. To mitigate the risks, we need to identify and analyze the risks that may occur before they cause damage on the project. At this point, the risk management plays a crucial role.

2 Description

This section consists of known and open risks that can occur in the project and the strategies to avoid them or the necessary actions if the risk occurs. The risks and strategies are listed below such that their importance decreases throughout the list.

1. The time required to develop the project is underestimated.

Strategy to be implemented: Create a plan like Kanban and if there is a delay, rearrange the development plan. Find some tutorials for a new technology if necessary. Prioritize the critical use cases.

2. Software tools cannot work together in an integrated way.

Strategy to be implemented: In order to easily eliminate any incompatibilities that may occur, a software tool plan describing their integration should be prepared.

3. Member leaves the group before the project finishes.

Strategy to be implemented: Reorganize the group such that the tasks of the leaving member are shared among the other members. Inform teaching assistants.

4. Changes occur in requirements that were not expected.

Strategy to be implemented: Derive traceability information to assess requirements change impact; maximize information hiding in the design.

5. Required training for group members is not available.

Strategy to be implemented: Members help and guide each other with their knowledge.

6. The organization is restructured so that different management are responsible for the project.

Strategy to be implemented: Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.

7. The customer requirements are not clearly defined and not completely understood by team members. The requirement specifications are ambiguous.

Strategy to be implemented: Clarify the requirements by communicating with teaching assistants.

8. A member is ill thus he/she cannot be available.

Strategy to be implemented: Reorganize the group such that there is more work to do for each member until the sick member heals.

9. Failure to implement wanted uses cases for the demo.

Strategy to be implemented: Inform teaching assistants about the failure. Represent the demo only with the implemented parts.

10. It is not possible to find a member with needed skills after a group member leaves the project.

Strategy to be implemented: Assign the tasks of the leaving group member, related with member's skills.

11. The specifications of crucial interfaces are not available on the schedule.

Strategy to be implemented: The specifications of crucial interfaces should be complete, consistent and verifiable by the determined time. If the conditions are unsatisfied, inform the customers about the situation and request an extension. Re-plan task shadings for specifications.

12. The database used in the system cannot process as many transactions per second as expected.

Strategy to be implemented: Investigate the possibility of buying a higher-performance database.

13. The size of the software is underestimated.

Strategy to be implemented: Write efficient code, if possible, use compression algorithms, increase the storage of the system to work.

14. The workforce of a group member is overestimated.

Strategy to be implemented: Redistribute the responsibilities among group members according to their work forces.

15. The rate of defect repair is underestimated.

Strategy to be implemented: Predefine enough repair time and solving mechanism that is within the software development schedule for every defect that may occur.

3 Risk Management Report Specifications

Risks forecasted in planning	How it was handled / mitigated
1. Some features can be too hard to implement.	Added features after building the base of the project and kept the project plan simple at the beginning.
2. Some software requirements cannot be thought beforehand. Discovering a requirement late can cause time and budget expansions.	Made a detailed research about the software tools and technologies and documented the results to reduce this risk.
3. Not being able to select proper architecture, missing some edge cases while thinking the test case scenarios.	Thought about all the possible cases together with every developer in the team to increase the test case numbers that are found.
4. High complexity in implementation. Using new technologies in the project can be hard and cost extra learning time.	Shared tasks with team members and taught the things that we learn will shorten the learning time.
5. The test that we have done can be insufficient in terms of different use cases. The documentation of the test results can be disorganized which causes confusion.	Recorded the test results in an organized way. Increased the scope of the test cases by analyzing the usages of the system.
6. Required training for group members is not available.	Members helped and guided each other with their knowledge.

7. The time and knowledge to combine the codes are underestimated.	More time wasted than the allotted time.
8. The work force of a group member is overestimated.	Redistribute the responsibilities among group members according to their work forces.
Risks NOT forecasted in planning, but observed	**How it was handled / mitigated**
1. Failure to implement wanted uses cases for the demo.	Informed teaching assistants about the failure. Represented the demo only with the implemented parts.
2. The time required to develop the project is underestimated.	Created a plan like Kanban and rearranged the development plan. Found some tutorials for the new technologies and discussed the solutions to the problems in the development process.
3. Unplanned work that must be executed	Reconsidered the architecture, assigned the work to the available developer.
4. Different technologies required to develop the system are needed.	Planned and organized new technologies, updated the architecture, and distributed the tasks to the developers.
5. Delays in earlier milestones / deliveries	Reassessed the project plan / release schedule, reorganized the developers accordingly.
6. The schedule of feature releasing was not in the proper order.	Rescheduled the timeline of releases.
7. Implementing new technologies cause complexity.	Refactored the system to reduce the complexity.

P.S: The red parts of the document are updated on 26.05.2021