HACETTEPE UNIVERSITY

Computer Engineering
BBM384 Software Engineering Laboratory

PepeFit
Risk Management Report

Group 10

EGE UÇAK	21527467
BAHADIR ADAK	21526619
EYÜP CAN BODUR	21526753
SERHAT SAĞLIK	21527285

PepeFit Risk Management Report

Risks forecasted in planning	**How it was handled / mitigated**
1. Incompletion of assigned tasks.	Using a reminder program.
2. Conflict problem when coding.	Using repository providing programs.
3. Wrong database technology choice.	Making research about database technology.
4. Product satisfactory problem.	We have to think realistically.
5. Ideas and limitations.	Simplicity and comfortability.
6. Correctness of our realistic requirements.	To talk with MacFit or who use sport center.
7. Confusion of system dependencies.	Use maven project and documentation.
8. Arrange work environment problem.	Define backup working environment.
Risks NOT forecasted in planning, but observed	**How it was handled / mitigated**
1. Get used to new technologies could be difficult.	Do a resource search with care.
2. Wrongly selected technologies and realized lately.	Talk with experienced people.
3. Storage of documentation may be disorganized.	Keeping our documentation in Asana, Git or Drive.
4. Customers may have problems when using the system.	Help tab can be added to the user interface.
5. Idea divergence in our team members.	Voting union.
6. Team members could be leaving team.	Find someone as Freelance or ask for help from other friends.
7. GitHub may be close again.	Keep backups of our work in a separate local place.

We mentioned listed above in our previous risk management report in DEL#3. We have faced some of these risks up to now. Short narrative descriptions are as follows:

Conflict problem when coding: Everyone who generates code on our team has its own branch. Apart from this, we have the main branch where we collect all the codes. After every major step in the project, we merge all our branches under the main branch. When we are doing this we often get a conflict but we are lucky that we think of this risk solution before. GitHub has a system that shows where users code has conflict in which line and who commit that code.

Wrong database technology choice: In fact we believe that we have chosen the right database technology but in the presentation of our project we received feedback that our database is slow. This is because we are students and we do not have financial support. Of course we found faster technologies in our research but they were charged. The best of the possibilities at hand and free one was the technology we used in the current system.

Get used to new technologies could be difficult: We have faced this risk many times; even one of our team members abandoned us because he could not cope with this risk. But as other team members, we have solved this problem by asking for help whenever we are in contact with each other or asking when there is a problem.

Idea divergence in our team members: In our team meetings, we talked about the progress of the project in general and decided what to do next week. There have been times when we have fallen into disagreement as to which priority should be given to when demo delivery of the project is approaching. In such cases, we voted within the team and made joint decisions.

Team members could leaving team: Because of what we mentioned in the previous risk, a team of friends abandoned us. We cannot follow the path of solution so we did not find someone as Freelance or ask for help from other friends. At first we thought our project work would get harder but then realized teammate who left, already has little contribution to the project and commit number is too low. We made a division of tasks again and finished the project with 4 people. After all, the risk is solved.

Our team experienced the risks mentioned above. In general, these risks did not have a major impact on the timing of our project because we thought that risk solutions before. We tried not to break the schedule of our project as much as we could. We submitted all demos and deliveries on time. At the beginning of the project we created class diagram and use cases. During the life time of project we tried to apply these UML components as much as possible but in some cases we were aware that some of them were wrong. When we look at the current another projects, the number of projects using that the waterfall model is already very low. Besides it is quite normal to have such changes because we cannot predict the future.

We will discuss the reasons why our team is not too much affected by the risks. At the beginning of the cause, it can be said that planning and scheduling is perfect because we use such system named Asana. Our project manager placed team duties in the most appropriate way to team member, when doing this he considered the knowledge and interests of the team members. Secondly, since we took front-end and back-end work parallel to each other, a MVC structure was formed in a short time. You can easily understand this from the demos we have shown before. Lastly, teamwork and hard work have made us achieve our success.

We have tried to follow as possible as to the plan of the project and to those mentioned in the previous reports. We actually encountered everything we planned ahead of time, so we had to put all the risks. Other possible risks that we may additionally indicate in addition to these risks:

Risks forecasted in planning	**How it was handled / mitigated**
1. User use the system incorrect way.	Prepare the user's manual.
2. User can forget his/her password.	Improve the forget password part.
3. User may want to provide feedback.	Put the contact part under the page.
4. System slowdown because of data traffic.	Speed up the database.
5. System security.	Measures taken against the attacks.
Risks NOT forecasted in planning, but observed	**How it was handled / mitigated**
1. Capacities of the courses reach the limit.	Request an extra course from sport center.
2. Inadequate number of trainers.	Hiring more trainers.
3. Mistakes during member registration.	System contains update operation in admin panel.
4. Server that the system is connected to may crash.	Reboot and maintain the server.
5. Think twice when we are accepting new team member.	We ask a question to new member for his/her about responsibilities.

User use the system incorrect way: In fact, the projected system is not so complicated but in some cases the user's mind will be confused. Prevent such situations and resolve this risk, user's manual can be created. This manual will include possible problems in this and frequently asked questions.

User can forget his/her password: The user is prompted for a password when logging in to our system. In some cases, the user may forget his/her password. In such risk situations, we need to develop the forget password section to help the user in the best way possible. At the same time, we should pay attention to the security and privacy of user information.

User may want to provide feedback: The user may want to share his/her observations with developers after using the system. Leaving such opportunities creates a risk for us because our system may become better with positive or negative feedback like this. This risk’s solution was designed to allow users to access whenever they want by putting contact under each page.

System slowdown because of data traffic: The backend of our system has a dataflow because the system we are using is sourced from the database. In some cases this data traffic can be intensive and this causes access to system be slow. To prevent such risks, we need to move our database technology to higher levels for speed up but this can increase costs.

System security: Today's many web based systems are very open against cyber-attacks. The use of online banking systems in these systems can even encourage cyber-attacks. When we think about all of this, it has to be evaluated in case of a risk against our own system. We are fortunate that security packages against this attack are also available on the market also our database in protection against attacks.

Inadequate number of trainers: If our gym becomes popular, the number of members will also increase. Increasing number of members will also increase the demand for trainers. The number of trainees will be less in case of such a risk. As prevention, we must keep the number of trainers up-to-date with supply and demand.

Capacities of the courses reach the limit: When the number of members increases, our courses will not be enough for our members. Our system already shows how many people are registered to the course; if the course is full already the system does not show that course. But customer satisfaction must always find the course the member is looking for. When we take this risk, we must keep the number of courses constantly up to date.

Mistakes during member registration: When one member wants to register the system, our employee doing registration. Incorrect entries during this process may create a risk. It can be 2 ways to get the risk. The first update can be done directly from admin panel. The second can be accessed and modified directly from the database, but the latter is not preferred for security reasons.

Server that the system is connected to may crash: Our system has server which name is Google Cloud Server. In some cases our server will be shut down, we have to consider that risk and make back up plan for that situation for solution we have to quickly reboot the system then in available time maintain the system.

Think twice when we are accepting new team member: Our tester in the middle of the project left us because he had trouble getting used to new technologies. This is an experience to us and we have to take this risk into account which chooses team member carefully in next project. As a solution, we need to evaluate the responsibility of the new candidate team member.