

<u>Risks forecasted in planning</u>	<u>How it was handled / mitigated</u>
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.
<u>Risks NOT forecasted in planning, but observed</u>	<u>How it was handled / mitigated</u>
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.

Incompletion of assigned tasks: Everyone in the project has a different schedule so there may be those who are unable to fulfill assigned duties. As we use Asana, the notification system is notified when the tasks are close to completion to solve that or we can ask the project manager to change the task.

Conflict problem when coding: We are collaborating in that project, there may be conflicts when we are working on the same class in the same section and when programming. We are thinking of using Git to solve this like creating branch.

Wrong database technology choice: If we do not choose the database technology wisely we will use later, the information of our users and trainers may be lost or we get the wrong result from the database system. Prevent to this making our research about database technologies more careful.

Product satisfactory problem: The product we produce will not satisfy requirement of members or trainers which defined before start the project so we have to think realistically and decide what we can do and what we cannot do when building project to avoid that.

Ideas and limitations: It may not be easy to put the ideas that we plan to do, into the web application as code. An idea that emerges can be creative but in the coding phase this can force us so we have to design an application that is as simple and comfortable as it can be. They are also provides convenience for the member or trainer when using application.

Correctness of our realistic requirements: We cannot be sure about the correctness of our realistic requirements because none of our team member used system like we are working on project in that reason we talked with MacFit to understand what a system really is and asked people who have used this system before.

Confusion of system dependencies: Libraries, dependencies can be confused or forgotten because we are working on MVC pattern and in that pattern we will see many components and technologies. Solution of that is planning to work by creating a maven project and prepare documentation for other members of group.

Arrange work environment problem: Work environment where we organize group meetings may not always be available. We usually organize group meetings in the study rooms in the library of our university but we cannot always find empty study rooms to prevent that determining the working environment in which we hold backup for group meetings like working environment of our department before.

Get used to new technologies could be difficult: We all have a coding knowledge, but none of us in our team have already designed a web application. So it can be difficult to learn new technologies. To take care that the resources to be investigated are as clear and as simple as possible then add them to the archive which purpose that share other team member.

Wrongly selected technologies and realized lately: We may regret that already used technology at the middle of project because we do not know all things in the beginning of the project. Not to return back and change system because we are using waterfall model so to solve this problem and choose best technologies, we got their ideas in consultation with experienced people like our teachers, teaching assistants or friends.

Storage of documentation may be disorganized: All of us will be responsible for specific part of project, but everyone in the project needs to know how things work. This can make the documentation confusing and disorganized because each team member's note-taking style can be different. To solve that, we can reach when we want by keeping our documentation in Asana or Git by the names and subject titles of team members.

Customers may have problems when using the system: We are trying to design system as much as possible simply and handle comfortable but it does not mean that users using this system will have no problems. 'Help' menu tab will be added to the user interface so that users can solve problems when they are having problems using system.

Idea divergence in our team members: In terms of task distribution or in other matters cause to be divergence the idea. For example some of team members wants to solve in different way and other member do not agree with their solution. In this case best solution is a joint decision must be reached with the voting union.

Team members could leaving team: Our friends who work in the project may have to leave the project due to their health condition or other personal reasons. In that situation do not disrupt the schedule progress find someone as freelance or ask for help from other friends which they are not involved other course projects.

GitHub may be close again: It has already happen in October 9, 2016. Our project source codes and diagrams store in the GitHub. Against the possibility of reclose of GitHub we take precaution like keep backups of our work in a separate place and personal computer.

Risk analysis has an extremely important place in the progress of the project and project's future. To predict and comment about the risks that may arise before starting a project and it is easier to find a solution when that risk occurs at the middle of project plan because if we are aware of the risks that may arise at the beginning of the project and if we prepare ourselves for it, we can easily get over it when the risk becomes a problem. It is worth mentioning that, there is no certainty that the above mentioned risks will be occurs.

If we will talk about the risks in our project; the most significant and results are most influential risk would be to break the schedule which planned at the beginning of the project. We tried to fit into our exam and assignments calendar as much as possible before we started the project. If we break the scheduling line several times, we were aware that the delivery of our project would be delayed. Other risks also as important as this one but other risks are less likely to occur. It does not mean we should not take this into consideration.

We will classify the risks listed above. We can say technical risks to the risks associated with the technologies, documentation, libraries used in our project. Miss the schedule line and health problems are included in direct risk category because their effect on the project is excessive. Incorrect timing planning or wrong organization, incapability of proper working environment these are related to resource risk category. After finish the project, if we will produced an application that cannot satisfy the specified requirement which defined before start the project risk is business risk.

The risks may have a positive or negative effect on our project. If we talk about these positive or negative effects for example, we face to face a risk and that impact our project in middle level then it may cause to earlier noticed that fatal risks that may arise in the future. Such a risk can also be seen as an opportunity because it aware us before see that fatal risk. On the other hand, we can put risks into the negative risk category that can deeply affect our project. For example, our database system is completely erasing an error result or misses the deadline of delivery time due to our inconsistency in the schedule timing.

As a result we can say that even thinking of the risk scenarios occurrence initially frightens all team members but if we can manage these risks in efficient way, we can translate this to our benefit. When we turn to benefit, contribution to the project is perfect.