**Risk: Changes in Requirements**

**· Mitigation**

In order to prevent this, we should analyze requirement properly, and conduct meeting with team member on regular basis.

In our project , chances of change in requirement will be less as we don’t have specific client to deal with. And we have decided what we will going to deliver at end.

**· Monitoring**

The meetings with the Team should ensure that our team members understand each other and the requirements for the project.

**· Management**

We are using the Github (version control tool) for managing any changes.

**Risk: Lack of Development Experience**

**· Mitigation**

In order to prevent this from happening, the development team will be required to learn the languages and techniques necessary to develop this software. The member of the team that is the most experienced in a particular facet of the development tools will need to instruct those who are not as well versed.

**· Monitoring**

Each member of the team should watch and see areas where another team member may be weak. Also if one of the members is weak in a particular area it should be brought to the attention by that member, to the other members.

**· Management**

The members who have the most experience in a particular area will be required to help those who don’t out should it come to the attention of the team that a particular member needs help.

**Risk: Database is not Stable**

**· Mitigation**

In order to prevent this from happening, developers who are in contact with the database, and/or use functions that interact with the database, should keep in mind the possible errors that could be caused due to poor programming/error checking.

These issues should be brought to the attention of each of the other members that are also in contact with the database.

**· Monitoring**

Each user should be sure that the database is left in the condition it was before it was touched, to identify possible problems. The first notice of database errors should be brought to the attention of the other team members.

**· Management**

Should this occur, the team would call a meeting and discuss the causes of the database instability, along with possible solutions.

**Risk: Poor Quality Documentation**

**· Mitigation**

In order to prevent this from happening, members who are in charge of developing the documentation will keep in contact with each developer on the team. Meetings will be held routinely to offer documentation suggestions and topics. Any topic deemed missing by a particular developer will be discussed and it will be decided whether or not to add that particular topic to the documentation. In addition, beta testers will be questioned about their opinion of the documentation.

**· Monitoring**

Throughout development or normal in and out of house testing, the development team and or beta testers will need to keep their eyes open for any possible documentation topics that have not been included.

**· Management**

Should this occur, the team would call a meeting and discuss the addition of new topics, or removal of unnecessary topics into the documentation.