**CHAPTER 3**

**CONSIDERATIONS**

1. **Performance**

Jewelry Palace requires a very low bandwidth, henge the performance will not degrade with increasing number of potential users. At the development stage, a free hosting service will be used. But when installing the system to a real university environment, it will be hosted in a much more reliable server to increase the performance.

MySQL will provide the adequate speed for database transactions. Since no big data analysis is done, MySQL is the ideal database for this project.

- Response time: less than 2 seconds

- Processing time: less than 2 seconds (no batch processing involved)

- Query and reporting times: yet to be tested

- Throughput: yet to be tested

- Storage: yet to be tested

1. **Security**

Security measures are provided in many aspects in this system. Users will have to authenticate using the username and passwords. If the user has entered a wrong name and passwords, they will not be able to login. If the password has been forgotten, the user can also show and change the forget password. Depending on the user, each user will have the functionality of the system. Each user's login time and logout time will be recorded in the system, to make the tractability process easy in case of a faulty action.

1. **Usability and Ease of Use**

The user is to provide with a complete user manual as a pdf file. The interfaces are designed to be easy and familiar to any potential user. No additional training is required to use the system.

* 1. **Capacity and Scalability**

Jewelry Palace can handle increasing traffic and scale effectively to meet the demands of users.

1. **Availability**

Jewelry Palace will be available throughout the 24 hours. Mean time to failure and mean time to repair will be decided to increase the availability. With a paid hosting space, the availability can be guaranteed to a great precision.

1. **Maintainability**

Jewelry Palace is designed the best practices of Java EE. This can be managed, updated, and enhanced over time. Here are some strategies to achieve maintainability:

* Automated Testing
* Code Quality and Standards