Why is the project important?

The software which will be developed for this project will concentrate on increasing the customer satisfaction after purchasing from a shop. Lots of users will be happy to use this software because it provides them the opportunity to gain benefits from their purchases. In other words, the development of such a software will help people spend less money on their everyday buying. For example, the software will be able to provide accurate suggestion of items to the users which will consist of the most frequently bought items.

The importance of this project is not only the fact that the potential customers can spend less money on their everyday purchases. After the initial literature review regarding the topic of “online shopping” and “smart shopping cart apps” it was evident that such applications do not satisfy the needs of the everyday customer or their functionalities are just limited. Furthermore, it can be clearly seen from the following paragraphs that there are not that much good web applications which provide the opportunity for their users to see a suggestion for the cheapest location from where he/she can buy the desired items.

Apart from the given examples of the importance of the development of such product, it can be clearly stated that this piece of software will be prompted to upgrading. In other words, new functionalities can be implemented in it in the future. Such functionalities may make a significant change in the habits of shopping of our users such as: less queueing time, faster decision to buy a good, less money spend on unnecessary items.

Finally, after the end of the literature review part of this assignment, it was more than obvious that the already created products provide limited functionality on group shopping. In other words, the existing software do not support fully the purchases of specific products and the chance for the users to split the prize with different users. The software which will be created will provide its users to buy a specific item and to share the price with different users. Apart from this function our users will be given a percentage off (discount) for the most frequently bought items.

Break down the project

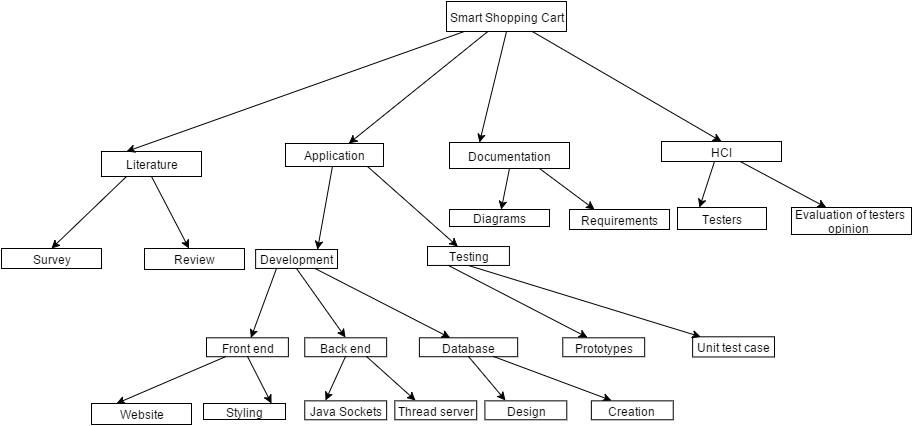
Before the creation of the actual code a valid architecture type should be chosen. Probably the most obvious choice of an architecture style which can be used for an application of this nature is the three layered Client-Server architecture style. This means that the software can be divided easily into three main layer

* The client part. This layer of the software will concentrate entirely on the user interaction and the user experience. This means that a very minimalistic and intuitive user interface will be created using several web technologies such as JavaScript and HTML which will ensure the user that their experience using the software will be pleasant.
* Server layer. This layer will concentrate mainly on the connection between the client and the server. Furthermore by using Java thread programming a way of communicating between the client and the server will be established. In other words, this layer of the software will create a way for the server to respond to various client requests.
* Database layer this layer of the software will be its storage. This means that the user information will be stored in the database in an encrypted way. The database will be designed in such a way which will guarantee the user that their personal information will be kept secret and well organized. Furthermore the good design of this layer will ensure that data will not be corrupted nor modified

However, it could be clearly stated that the creation of such a software is not the only requirement which must be met in order the project to be with a good quality. There following bullet points will list the additional steps which must be taken.

* **Documentation:** Apart from the creation of an intuitive way for presenting the functionalities of the software the creation of well and clearly written user documentation is a good step which could be taken towards the delivery of good user experience. Furthermore this will make the user`s time to get used to interact with the program less.
* **Testing:** Probably the most efficient way to see whether users like the product is to give them the opportunity to try it. Towards the course of this project users will be asked to try the functionalities implemented in the software. Their opinion will be written down and taken into consideration. This approach will help me to eliminate errors or bugs which might be found by the testers.

The following diagram will present the way that the project will be broken down into sub topics.



**Risks**

As seen from the task description and the module booklet (check it xD) this task is closely related to good time management. This means that the main risk which this project will hide Is the incorrect time management or spending more time on solving a particular task. The following paragraph will contain a list of the functionalities which might be more time consuming

* The collection of valuable user information will cause time and it is not guaranteed that the opinion of the user will contribute towards the eliminations of problems.
* The results from the testers` feedbacks might be unhelpful.

The following approaches will be used to eliminate to ensure that regardless the presented risks the project will be successfully created in the given time.

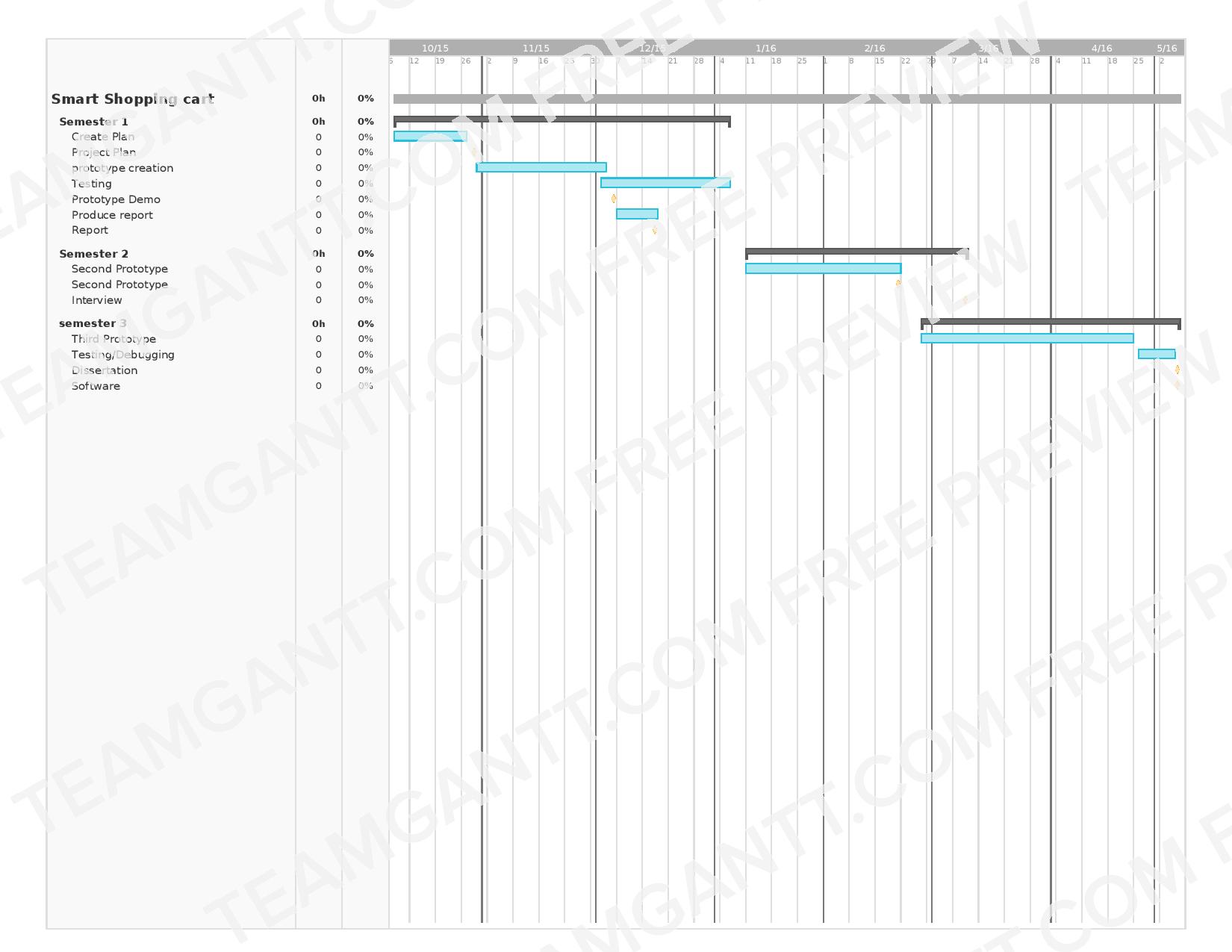
* The users who will test the product will be given a questionnaire with clearly presented questions and with clear answers. This will improve the amount of helpful information which will be given by the testers. Also, the creation of such a “template” will help our testers to focus more on the vulnerable or not that well developed aspects of the program.
* Different groups of testers will be invited. This means that apart from everyday users, people with some experience in programming will participate. This will ensure that the results of the questionnaire will be helpful.

**Milestones**

There will be two different types of milestones which will be created for this project.

* Internal milestones- The given deadlines for this module will be used as an internal milestones which will track the creation and the delivery of the deliverables in the required time
* Prototype milestones – These milestones will be connected entirely to the functionality of the created software. They will show dates when different percentage of the entire list of functionalities must be met.
* First prototype – This prototype will implement couple of functionalities which will be shown during the prototype internal milestone.
* Second prototype – this prototype will cover 50% of the needed functionalities this prototype will be used the testers.
* Final prototype – this will be the final prototype which will be created. It will cover all of the necessary functionalities and will help towards the creation of error free stable version of the software.

**Gantt chart**

****

**Activities**

During this step of the assignment the breakdown of the project was taken into consideration and the following activities have been created. The duration of these activities is approximate so it is possible for variations of the durations to occur.

|  |  |
| --- | --- |
| **Activity Name** | **Duration** |
| **Literature Study** | **1 week** |
| **Literature review** | **2 weeks** |
| **Client** | **7 weeks** |
| **Server** | **7 weeks** |
| **Database** | **7 weeks** |
| **Analysing user reports** | **2 weeks** |
| **Validation** | **7 weeks** |

**Sub tasks**

Some of the listed activities in the presented table can involve the creation of smaller tasks. The following list of subtasks shows the activities which can have subtasks and shows the approximate time that will be needed for these tasks.

Database (6 weeks)

* Design (2 weeks)
* Implementation (4 weeks)

Analyzing user reports (3 weeks)

* Collection of reports (1 weeks)
* Analyzing the report (1 week)

Validation (7 weeks)

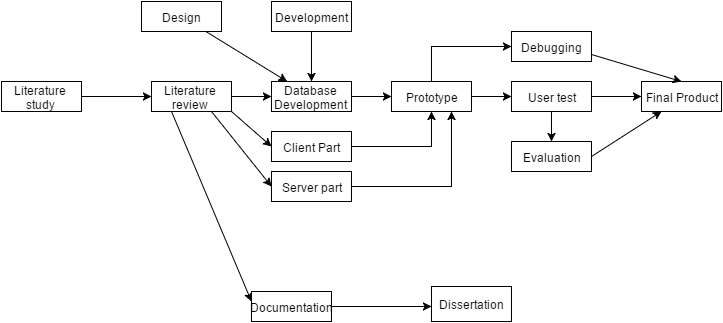
* Testing (2 weeks)
* Debugging the parts that were mentioned as “not entirely functional” by users (5 weeks)

Server (7 weeks)

* Design (3 weeks)
* Creation (4 weeks)

**Sequence of activities**

Based on the previous paragraph it can be clearly seen that the given set of activities will produce the following sequence of activities diagram.

****