Software Engineering

Software model report

Team Members:
Abhijeet Verma (B19CSE001)
Adityaraje Ashok Devade (B19CSE005)

Course Project : Inventory Management Application with focus on institute sports complex

Project description

This **project aims** to automate the process of issuing/returning and check availability of items from inventory , monitoring student dues or damaged items and keeping a record of inventory items used by sports facilities such as training centers, gymnasiums, stadiums, sports federations,universities or hostels.

Technologies decided to use : c++/c for writing code in modules , mysql for relational database stored on cloud server , mysql c api 5.6 for calling cloud sql database in c++/c application , cmake for building the project code into a executable that could be run in terminal .

Which Type of software process model was followed and why ?

The waterfall model is a sequential software development life cycle that is simple to understand and use. There are **in total 6 phases** in this model namely : requirement gathering and analysis ,system design , implementation ,testing ,deployment , maintenance. Each phase has to be completely finished before another start which means no overlapping is allowed. The output of each phase serves as the input for the next stage.

The waterfall model seemed to be suitable for the project due to the following reasons :

1. Since the project is to create a inventory management application with additional features of issuing, returning items and paying dues functionality . If extensive requirement analysis is done with client (institute administration) and other stakeholders like students ,

there is very less or no scope of requirement changes in between the project. So **the requirements are very well known in advance, clear and not supposed to change in future. Also the product definition is stable**

2. Since the coding stage is in C/C++ hence implementation phase would be easy to do if we have extensive design document with class diagrams , sequential diagrams , activity diagrams as well as a well defined SRS with usecases . Also the intended stakeholders/client are institute /hostel / sports society administration so to convey them the usability of the application , extensive documentation with well defined use cases / user scenarios is a plus point . **Therefore , the software needs well documented documentations - SRS , design architecture document , readme**
3. The technologies used for the application - C++/C , Cmake , Mysql , Mysql C api - are **mature technologies** that is - they have been in use for a long time and hence there are no unexpected risks and ample resources with required expertise available freely on internet . So its less likely that a bug would go unresolved and affect deadlines of a phase.
4. **The project is a small ,short and simple** one so it would not require complex SDLC models like iterative , agile , RAD etc .
5. Since the client is college administration therefore , **constant involvement of client in this project would not be possible** . intense Discussion could only be made at requirement gathering stage so there is no meaning to do iterative and other models
6. There is no much dependency on the end users students and administration during the development and other intermediate phases
7. Since this is a academic project , therefore **deadlines are crucial** . Therefore waterfall model , which is Easier to manage as there is clear schedule for each stage that gives clear milestones, would be more suitable
8. Constane measure of project progress is not required . progress can be tracked / reviewed at end of each phase or at the end of each week .

SDLC Different phase description

Conception ,Requirement gathering and analysis

The project idea for conceived due to the following problem which served as a opportunity to deliver a software solution :
Currently, in our and many other institutes, issuing of sports equipment by students is done manually and details are stored in registers. There is no automation in the process which leads to inefficiencies. Students sometimes bypass and do not register themselves before issuing or guards misplace the registers storing the issuing details. Also, no strict action is taken for damaging/misplacing equipment. Availability of a secured sports inventory management software would allow automating tasks like adding dues if

returned late or fine if the equipment is damaged, fast retrieval of information like inventory details, issuing details, and adding features like allow issuing only if students have no dues.

To get indepth idea of the problem and what the stakeholders (students , college administration , security persons) would want , interview and discussion was done with the stakeholders with several solutions proposed and discussed . The output of the discussion was a rough idea of the user requirements and quality attributes to consider for the software . Following points were noteworthy :

1. Each inventory item should be uniquely identified by a Id and items should be categorized based on their type . i.e basketball is a category and each 3 basketballs in inventory has their own id .
2. Issuing details like which item , by which student , issue date ad return date should be stored . issuing not allowed if student has any dues
3. Returning of item should come with automatic due addition to the student if he returns the item late.
4. There should be security that is student should not be able to change \ manipulate databases on their will.

The application is feasible as it could be develop in less time with minimal cost and serve a solution to a problem as identified by the stakeholders.

Defining requirements

Once the requirement analysis is done the next step is to clearly define and document the application requirements and get them approved from the client i.e institute administration. This is done through an SRS (Software Requirement Specification) document which consists of all the application requirements (scope , constraints , assumptions,hardware requirements , software requirements ,performance requirements , functional requirements , usecases/user scenarios, non - functional requirements ,schedule , budget) to be designed and developed during the project life cycle.

Design : design architecture document

Using the SRS from the previous step , the software object oriented design is defined using UML diagrams presented as a 4+1 view architecture .
In this phase , the structure is divided in modules with each module performing a functionality and serving a user requirement . The structure diagrams : component diagram and class diagram

defines the structure of code to be written and hence make the coding part easier whereas the behaviour diagrams - use case diagram , sequential diagram , activity diagram - defines the logic of each module . All these diagrams are formulated and presented as a 4+1 view design architecture document .

Implementation/Coding

Implementation Phase is where the actual code is written. Development of the software is done according to the instructions recorded on the documents prepared in SRS and Design documents .So not much decisions are to be made in this phase . The output is a final coded software which performs all the functionalities and is ready for testing .

Testing

Manual unit testing and black box testing is done for the final application . In manual unit testing , each functionality /module is tested and compared with expected output . In black box testing , the whole application is assumed to be a black box and inputs are given and compared with expected output defined by system specification

Deploying and Maintenance

Since it is a terminal executable application therefore , no such maintenance is required once the executable along with all the required files is installed on the computer . The only maintenance that is required is to maintain the sql cloud server which the application uses to store the database . If any bugs are reported by users/client , the code would be corrected and new zip folder with the executable will be given to the client with modifications . hence The client is provided with regular maintenance and support for the developed software.