DELIVERABLE 2 and 3– PROJECT PLAN

**Demo Functionality (Developer)**

PLEASE VIEW LECTURE 8 – DISCUSSES ABOUT GUI

PLEASE VIEW PROJECT BINDER FOR GUI LAYOUT & FUNCTIONALITY:

(Deliverable 1 -> Design -> Question 3 -> GUI User Navigation)

**Developers’ Task:**

* Aymen -> Login, Sign-up, User Management, Stock Control, Sales Control
* Rati -> Add User, Change Password
* Ethan -> Welcome, Refund, Sales Report
* Jacob -> Alerts, Stock Control, Sales Control, Sales Report

**GUI Forms:**

* Login
* Sign-up
* Change Password
* Welcome
* **User Management**
  + Only admin roles (Manager, Admin, Advisor) can: *(lecture 8 – page 32)*
    - Edit customer details
    - View general reporting of customers
    - Add discounts for valued customers
* Add User
* **Stock Control** *(lecture 8 – page 29)*
  + Adding and removing **blanks** from a **travel agent** (company)
  + Searching for **blanks**
  + Stock Turnover **report**
  + Assigning, removing, and reassigning **blanks** to a **travel advisor**
* **Sales Control** *(lecture 8 – page 30)*
  + Making a **sale** as an **advisor**
    - Different **customer types**
    - Different **sale types** (interline and domestic)
    - Different **payment types**
  + Recording **payments**
  + Refunding **tickets**
* **Sales Reports** *(lecture 8 – page 31)*
  + For or by an **advisor**
  + For the whole **agency**
  + Consider the different **reports** and calculations required etc.
  + Consider who should be able to produce the **reports**
* **Alerts** *(lecture 8 – page 33)*
* Refund

**Implementation Report (Tester)**

*PLEASE VIEW LECTURE 5 – IT DISCUSSES ABOUT IMPLEMENTATION REPORT IN MORE DETAILS*

*PLEASE VIEW OOAD LECTURE 9-IT GOES INTO DETAIL ABOUT HOW TO IMPLEMENT THE TESTS*

Testers’ Task:

* Abbib -> Writing implementation report, testing plan and testing (Customer package code and travel agent package code)
* Uzair ->Testing (Ticket package and report package)
* Ishmail ->Testing (SQL package)

Marking: *(lecture 5, page 45)*

Implementation Report [20 marks]

* Software architecture/ Compilation/ Run-time components [8 marks]
* Testing plans and reports [12 marks]

Structure: *(lecture 5, page 42)*

* Needed for product maintenance
* Failure to supply this will result in ZERO marks being awarded for the whole of the Implementation!
* Must include the following:
  1. Compilation *(lecture 5, page 47)*
     + What are the source files, programming language and compiler used (version, compiling and linking options, script files, e.g. main files). If applicable, provide a project file description (forms, data modules, etc.)
     + A list of created binary files (.exe + .dll or .jar files for Java):
       - their short description and where they should be located at run time (directories on the target machine). • If more than one .exe file is created (e.g. main file + backup/restore utility), a clear description should be given of who, and how, is going to use them.
  2. Appendix *(lecture 5, page 47)*
     + Commented source code files (in a zipped file for the online submission)
  3. Run time – Testing *(lecture 5, page 48)*
     + As a minimum, a list of the COTS (commercial off the shelf) software components that must be provided (static libraries, DLLs, etc.) which the application needs to run properly.
     + Dependencies between components (DLLs, etc.) to be shown, e.g. using UML component diagrams.
     + Deployment of the run time files on the target computer(s)
  4. A testing plan for (as defined above) and testing results *(lecture 5, page 48)*
  5. Documentation *(lecture 5, page 48)*
     + Technical documentation of your codebase and database as required
     + User documentation to aid new users to learn the system

Things to do: *(lecture 5, page 43, 51-56)*

* Define the programming language and develop the codebase
* Develop the database in MySQL
* Test and refine
* Optional: Add UML Component / Deployment Diagram
* Present and demonstrate your final software product

Preparation: *(lecture 5, page 44)*

* This deliverable must be a single, well formatted and presented report
* Insert your diagrams into the document to preserve your page numbering etc.
* Ensure all diagrams are an appropriate size (readable)
* You may decide to use UML Component (or even Deployment) diagrams in the report, but this is not mandatory. Textual description is also acceptable, provided it is complete and correct

**Weekly Plan / Gantt Chart**

**Developers:**

*(DUE Thursday, April 13th)*

* Week 9 (20/03/23 – 26/03/23):
  + Each member will work on their chosen GUI form
  + Due to us having so many coursework, it’s best to start early from now and do at least 30 minutes DAILY on our task.
  + Functionality is main priority. (Example: Making sure buttons works, adding text fields, etc.)
  + Any changes that you make on the code – please push it onto GitHub
* Week 10 (27/03/23 – 02/04/23):
  + Databases (Example: Creating a new account will produce a database of emails and password, reports, etc.)
  + JavaDoc (discussed in lecture 8)
  + Will be discussed further soon on what we should do
* Week 11 (03/04/23 – 09/04/23):
  + Final code must be completed
  + Make the GUI look appealing and professional!
  + Let the tester know that the program is completed
  + Submit the coursework!

**Testers:**

*(DUE Sunday, April 16th)*

* Week 9 (27/03/2023-02/04/2023):
* Set up the Implementation report.
* Revising OOAD lecture 9 for testing.
* Writing testing plan
* Revising OOAD lecture 8 for UML deployment diagram.
* Maybe writing the user documentation since we already know what the system should have?
* Week 10 (03/04/2023-09/04/2023):
* Started testing our allocated packages as soon as they are implemented.
* Publishing the test results
* Finishing the technical documentation for our implementation report
* Finishing the deployment diagram
* Week 11(10/04/2023-16/04/2023):
* Inputting the compilation details into the implementation report
* Inputting run time dependency details and deployment diagram into the implementation report
* Inputting the commented source code into the submission
* Checking over the implementation report
* Submitting the implementation report!

|  |  |  |  |
| --- | --- | --- | --- |
|  | Week 9 | Week 10 | Week 11 |
| GUI |  |  |  |
| Code implementation |  |  |  |
| Database |  |  |  |
| Testing plan |  |  |  |
| Unit/use base testing |  |  |  |
| Implementation report document |  |  |  |
| Submit deliverable 2 |  |  |  |