CMP-5012B Project Stage 2:

Study Planner Implementation

The University of East Anglia (ie the client) has reviewed the design they have commissioned for the Study Planner and discussed that with their key developers and stakeholders. This review has resulted in some clarifications and amendments. On the basis of these, and the background information that was provided to the design team, you are now given the task of reviewing your design, and then to implement a prototype system. The amendments and clarifications are:

- The system is to be based on the Java technology stack, i.e. it specifically must use JAVA classes.
- Design documentation must be provided, which includes UML diagrams and a complete description of the design phases, assumptions and design decisions that have been made.
- The system must be implemented as a stand-alone desktop application
- You must employ a Model-View-Controller architecture for your application
- You have to perform two implementation iterations, both of which need to be documented clearly in your report and documentation
- The client is keen that the new system should be scalable and provide good performance under high loads. Therefore, they ask that for demonstration purposes, the development team creates a number of semester files that include modules, coursework and deadlines
- The client also values extensibility of the new system as this will allow them to introduce new functionality in the future. Based on the prioritisation below, the client will appreciate that when certain requirements cannot be achieved in the finished prototype, that they can easily be introduced into the system later on by means of extension mechanisms, such as inheritance and aggregation structures.
- The code for the study planner must be created by your team. The use of libraries is only allowed with explicit confirmation by the teaching team. The use of a library for displaying a GANTT chart is **not** allowed.

The client recognises that some of these revisions can be achieved in multiple ways and agree that the development team can change the design and database structure as necessary to facilitate scalability and performance. However, a core set of functionality needs to be provided. On the basis of the revisions and the original background information that was provided to the design team, you have agreed with the client on the following requirements in MoSCoW form:

Priority	**Requirement**
must	• Be able to load a semester file containing module, coursework and deadline information, and exam information from a defined file format • Be able to define study milestones which must be attached to coursework or exams • Be able to define study tasks contributing towards specific coursework or exams • Be able to store defined milestones and tasks on disk as part of the semester profile
should	• Be able to record study activities that are related to study tasks and contribute towards completing milestones • Be able to store activities in the semester profile file • Have a study progress dashboard that highlights upcoming deadlines, progress towards completing milestones and time spent for each module • Be able to visualise activities, dependencies, milestones and deadlines in a Gantt chart representation
could	• Be able to process updated semester files with new deadlines and updating the stored activities, tasks and deadlines • Be able to highlight the critical path of activities within the Gantt chart • Be able to visually highlight progress made in the Gant chart based on study activities
won't	• Support direct communication with the hub via internet • Download semester profile files from an external resource • Require hub or faculty staff members to interact with the system directly

Original Background on Study Planner (as Provided to the Design Team)

SCENARIO: Study Planner **VERSION: 1.0**

Author: Joost Noppen

Date: 19th December 2016

1. OVERVIEW

UEA is always looking for ways to support students in their studies and with many of the modules and assignments requiring detailed planning, prioritisation and tracking of tasks the university wants you to develop the "Study Planner" application. The goal of the Study Planner is to help students schedule tasks based on their module schedule and to define tasks and their dependencies for which they can then track their progress. Initially this application will be offered by the School of Computing Sciences but support for other schools in the future should be considered.

2. SOLUTION DESCRIPTION

The Study Planner application is a support tool that students can use to schedule and keep track of their study activities and to determine their progress towards completing their coursework assignments. For this the student has to be able to define record details of activities which includes start date, end date, module and coursework assignment it relates to, task progress and dependencies that can exist between tasks (such as one task can only be started after another has been completed). The application should be able to visualise the planned tasks and their dependencies by means of a Gantt chart representation. To monitor progress in a glance the application must support a Study Dashboard that highlights upcoming deadlines, progress towards milestones, etc.

The module, coursework and deadline information for modules taken by the students should be acquired by loading a file containing these details which will be provided by the HUB on an individual student basis. It is up to the design team to propose a file format for this and the details that the HUB will be expected to provide. If this initial experiment for the Study Planner is successful it can be further integrated with the HUB in the future by allowing the submission of coursework files and integrating news feeds with updates from module organisers.

Four central capabilities are required by the solution. The capability to:

1. Load module, coursework and deadline information from a defined file format
2. Ability to define study tasks, details, milestones and deadlines
3. Ability to record study activities that contribute towards completing study tasks and milestones
4. Visualise activities, dependencies, intermediate milestones and deadlines in a Gantt chart representation as well as a study progress dashboard that highlights upcoming deadlines, progress towards completing milestones and time spent for each module

The basic workflow of the Study Planner Application is outlined below:

Creating a Semester Study Profile

1. Initiating a new semester study profile
   a. The user creates a new study profile for the upcoming semester by clicking the appropriate button and provides an identifier for this semester (e.g. Spring Semester 2015-2016)
   b. Upon creation the system prompts the user for the semester file provided by the HUB. If an invalid data file is provided an error is raised and the user is prompted to provide a valid data file before allowing the creation of the semester study profile to be completed
   c. For each module listed in the data file the application creates a content area that will be used to display activities, milestones, deadlines, etc.
2. Inspecting the semester study profile
   a. The user is presented with an overview area where he/she can select the modules in the current semester profile
   b. Upon selecting one of the modules the user is taken to the detailed module page that lists activities, milestones, deadlines, Gantt charts, etc.

Plan Study Tasks and Milestones

1. After loading the data file the module profiles are populated with coursework assignments and exams, their weightings and their deadlines
2. The user can now start planning their semester by defining study tasks that contribute towards completing coursework assignments or preparing for an exam
3. Study tasks must belong to a specific assessment event of the module (coursework, exam) and the application must at least capture the time that will be spent, the type of task (studying, programming, writing, etc.). The type of task is a defined set of types
4. Each task must have a requirement criterion in it based on which progress and completion can be assessed, such as time studied, book chapters covered, assignment requirements completed, etc.
5. Tasks can have dependencies on each other, such as a task cannot be started before another has been completed. Other tasks can be done in parallel. The application must support this information in the tasks

6. Each task must have the possibility to add notes
7. In addition the user can define milestones (intermediate deliverables with their own deadline) that are part of completing coursework or preparing for an exam. Milestones must be related to the tasks that are required to be completed before the milestone is achieved
8. Note that deadlines can change as a result of the module organiser changing it or by applying for an extension. This must be supported by the application.

Capture Study Progress

1. Study progress is captured by defining study activities such as programming or writing.
2. For each activity a quantity must be captured that can be used to evaluate its contribution, such as having studied for three hours, created three coursework parts, etc.
3. Each activity must be attached to at least one task and the details of the activity contribute towards completing the task. For example if a task requires five hours of studying and a related activity list that the user has studied two hours, this means that in the task there are three hours of studying left
4. In addition for each activity the time spent must be captured
5. Each activity must have the possibility to add notes
6. An activity can be attached to multiple tasks and thereby contribute to the completion of both

Study Dashboard

1. To examine how the study for the semester is progressing the user can open the Study Dashboard
2. When the dashboard opens, the application checks whether any deadlines for modules are approaching or have passed
3. Next the application determines whether the tasks associated with the deadline have been completed (i.e. the activities associated with the task are sufficient to satisfy the task requirement)
    a. If all the tasks for a specific deadline have been completed it will be added to the deadlines section of the dashboard
    b. If not all tasks for a deadline have been completed and the deadline has not passed, it is added to the upcoming deadlines section of the dashboard
    c. Otherwise the deadline is added to the missed deadline section of the dashboard
4. For each deadline a progress bar must be displayed that highlights how far the user is along to completing the work
5. The Study Dashboard must allow for users to examine the individual module planning and tracking they have done by opening a Gantt chart representation of the data they have entered. This should both visualise the tasks, milestones and deadlines, as well as the activities and how this is contributing towards completing the tasks.