****

**An E-Learning App to Teach Programming**

**Description**

Learning to program can be difficult for first time users of computers and software. Each programming language has a set of key language commands and libraries that students need to learn. There is usually a development environment used to edit, execute and debug programs. Students often find it difficult to absorb all the materials that lecturers teach in class. Lecturers, on the other hand, are also unlikely to be able to go through each step of editing and debugging code.

The objective of this project is to develop an easy-to-use e-learning system for students to learn a programming language. Only one single language requires to be taught by the app. The system should allow its users to learn the background of a language, the key commands, how to write and execute a program. The students should be able to watch sample programs and how they through step-by-step demonstrations. Students will be provided with quizzes to test their understanding.

Lecturers should be able to design quizzes, analyse results of quizzes and include them as a part of module continuous assessments.

This system is intended for teaching and learning programming of a single programming language. It should allow students to write and execute simple program through an intuitive user interface.

\*Ensure that all material is referenced correctly.

**Requirements Engineering**

**Feasibility Study**

1. Identify at least 4 examples of E-Learning systems in the marketplace. Briefly describe each of them. Reference the URL for each. (in Windows, Android, iOS, …)? *(min 100 words)*
2. Identify the main the main system features and services provided in an e-Learning system. Consider the existing systems and the services they provide to users and lecturers.

[Reference and copy the URLs] *(min 200 words)*

1. Describe a new type of basic e-Learning system and how it might operate. Consider existing systems that provide services to users and lecturers. Use the internet to identify
2. In what ways do students currently learn to program? *(min 100 words)*
3. Who are the stakeholders? Would this app affect them positively or negatively? *(min 30 words)*
4. What other research would be necessary to ascertain feasibility e.g., ownership of smartphones/tablets, e-learning …? *(min 100 words)*
5. Make an initial list of **functional** and **non-functional** requirements.

**Requirements Elicitation**

1. Could ethnographic methods be used in this case study? If so, in what way? *(min 30 words)*
2. Identify a significant stakeholder(s), which will be **interview**ed to get more information on the intended product. Justify your choice of stakeholder(s). Do up an interview plan and pre-prepare approximately 10 questions.
3. Identify a significant group of stakeholders, which will receive **questionnaires**. Justify your choice of stakeholders. The questionnaire that you create should have approximately 10 questions.

Requirements Analysis

1. Use the use case template to analyse the proposed system

System Name

(Automated Library System)

Actor

(Student)

The use case description is developed from analysing the description of the use case. This is the statement of the goal of the use case.

For the first iteration this will be a description of the how the system operates.

Use Cases focus on functional requirements and specific system behaviour.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE** | | <number> | <Name of Use Case>  <the name is the goal as a short active verb phrase> | |
| **Description of Goal in Context** | | <a longer statement of the goal for this Use Case in context > | | |
| **Preconditions** | | <what we expect is already the state of the system>  <list> | | |
| **Post Conditions, Success End Condition** | | <the state of the system upon successful completion> | | |
| **DESCRIPTION** | | < The use case description is a paragraph identifying behaviour, it comes from the requirements gathering> | | |
| **Main Flow** | | | | |
| **Step** | **Action** | | | **Alternate** |
| n.1 | <put here the steps of the scenario from trigger to goal delivery, and any clean-up after> | | |  |
| n.2 | **<…>** | | |  |
| n.3 | **<…>** | | |  |
| n.4 | **<…>** | | |  |
| n.5 | **<…>** | | |  |
| n… | **<…>** | | |  |
|  | | | | |
| **EXCEPTIONS or ERROR Flow Description** | | | | |
| **Step** | **Branching Action**  < Exception number m of Use Case n> | | | **Alternate** |
| n.m.1 | < condition causing exception>  <Action, steps of scenario to goal delivery> | | |  |
| n.m.2 | < condition causing exception>  <Action, steps of scenario to goal delivery> | | |  |
| n.m.3 | < condition causing exception>  <Action, steps of scenario to goal delivery> | | |  |
| n.m.4 | < condition causing exception>  <Action, steps of scenario to goal delivery> | | |  |
|  | | | | |
| **ALTERNATIVE or VARIATION Flow Description** | | | | |
| **Step** | **Branching Action** | | | **Alternate** |
| n.m.1 | <condition causing alternative>  <list of variation>  <Action, steps of scenario to goal delivery> | | |  |
| n.m.2 | <Action, steps of scenario to goal delivery> | | |  |

Non-functional requirements, management issues and decisions required to be made, can be identified in the following table

|  |  |  |
| --- | --- | --- |
| **RELATED INFORMATION** | Use Case: <number> | <Use case name> |
| **Priority:** | <how critical to your system/organization> | |
| **Performance** | <the amount of time this use case should take> | |
| **Frequency** | <how often it is expected to happen> | |
| **Channels to actors** | <e.g. interactive, static files, database, timeouts> | |
| **OPEN ISSUES** | <list of issues awaiting decision affecting this use case> | |
| **Due Date** | <date or release needed> | |

**Requirements Specification**

1. From the initial requirements analysis identified using the use Case scenario analysis identify key functional requirements.

Fill up the following matrix with *functional* requirements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Req ID** | **Name of Req** | **Description** | **Priority** | **User Contact** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**System Modeling**

1. Describe the primary way that a user may use this application. Describe the system from the student view and then from the lecturer view. This will describe the Main flow and identify alternative flows for the overall behavior of the system.
2. Who are the possible *actors* in this system?
3. What are the possible *use-cases* in this system?
4. Draw an initial *use-case diagram* with supporting scenario description for this app (possibly using *StarUML* for the diagram). The first iteration of the use-case diagram can consist of a single overall use case with supporting main flow and 2 or 3 alternative flows. Insert the picture NOT the file into this document.