|  |
| --- |
| Team Members:Name Student Number Primary Role Secondary Role Ryan McDonnell C12460362 Team Leader  Jennifer Mullins C12479202 Tester  Fabio Dayrell Rosa D14128448 Documenter  Bilguun Nemekhbaatar C13329456 Planner  Matheus Connolyn D14128350 Coder  *Roles:*   * *Team lead – maintain this document allocate tasks* * *Developer* * *Tester* * *Documentation* |
| Project Title E – Learning Web Application Tool |
| Version Number: 4.0 **Change History:**   |  |  |  | | --- | --- | --- | | **Date** | **Author** | **Comments** | | **01/03/16** | **Jennifer** | **Version 1** | | **02/03/16** | **Fabio** | **Edited Version 2** | | **03/03/16** | **Matheus** | **Edited Version3** | | **04/03/16** | **Bilguun** | **Edited Final Version** | |
| Summary We will be creating an E-Learning Tool for first year Computer Science and Engineering students to aid their programming skills outside of college. The system is only accessible using a DIT verified email account. Our aim is to provide a fun alternative for revising and practising programming. Our tool will mainly focus on the C Language.  The tool will revolve around the idea of a quiz where users can compete against other students in a “ranked” mode where there are leader boards and high scores. We feel that competition is a great way for students to learn.  However the student has the option of choosing a “non-ranked” quiz in which all scores are kept private. It is important to note that the “ranked” and “non-ranked” modes will be completely optional and the user can do the “non-ranked” version where the results are kept private for only the user.  We wanted to implement a social aspect to the tool as well, giving the users a forum where they can post questions and answers and also add other students, check progress etc. The user will have a profile page where their personal information is stored such as their name, their high score and a progress bar. |
| Proposed Approach  1. **Overall approach**   One of the main risks of a team project is that due to the lack of architecture, everybody is moving in small steps and sometimes the full picture is lost. We will avoid this problem by setting clear goals and requirements each iteration and clear communication within the team. Since we are using the SCRUM methodology, we will be going through sprint cycles after identifying the requirements and we can change our approach according to the users and customer’s needs and wants.  Another main risk of Agile methodologies especially using SCRUM is the one step forward and two back problem, this is when you have met with the client and certain changes to the development mean changing a lot more than anticipated which can be frustrating and most important it is time consuming.  In order to tackle risks We will implement a risk register for risk management:  **Description of risk:** A one- or two-line overview of the risk. It should be simple and easy to comprehend.  **Likelihood:** Estimated probability of occurrence of the risk.  **Severity:** The severity of the risk is assessed based on impact of the outcome.  **Owner:** The person who manages, controls, and takes action in response to the risk.  **Action:** The response defined to manage/control the risk.  At each meeting, we will look over the risk register and manage and monitor the risk collectively as a team. It is important to keep a constant communication within the team.   1. **Requirements Phase:**   We will complete requirements gathering by many different ways but most importantly our goal is to work closely with the users. Ways in which we will do this is by using prototypes for the user to see the presentation of our development and gather essential feedback.  We will be using user stories.  A user story will describe the requirement at the highest level with a single feature with a regularized format. "As ‘X’ I want to ‘Y’ so I can ‘Z’".  For example:  "As a legally-blind user I can magnify my screen so I can see small screen elements".  We will then pile all the requirements into a sheet put more detailed description about the requirement. Then we will take all the stories  and put them in priority. Then the team will start working on the highest priority requirement and at the end of a sprint we show the users what we have achieved and either add new or take away requirements and repeat the process.  We will use appropriate use cases to show the user how they are will interact with our system and get feedback on this also.  We would also like to observe the user working with the systems prototypes such as using the application “Pencil” which is very useful for this kind of requirements gathering. We believe all the above will help reduce risk during this part of the development which will set us up for a very solid foundation. We believe that having all the essential requirements front his stage is going to reduce risk of failure **2. Design Phase:**  The method used for our system design is the scrum methodology which is an iterative agile software methodology. It’s good for our project because we work in close contact with the users and work in cycles. We will use Identify requirements and release a working version, receive feedback and repeat. We connect with the customer on a weekly basis which allows us to show our progress and take feedback so we can continuously add or take away from the design as we go along.  **3. Implementation Phase:**  The programming languages we will be using for the server-side will be PHP. PHP is the most used programming language for this purpose (web development), more than 80% of all the websites uses PHP (W3Techs - reference). For the database we are using MySQL and Apache as the web host. We will also use HTML5, CSS and JavaScript as these are the most popular languages used in web development and play a major part in producing content on a website.  All team members can work easily on code using GitHub, Also, using GitHub we have an efficient tool to version control. This will help reduce risk of losing source code and work. It also allows for member to work away from each other. To avoid risks during implementation we will plan early, communicate effectively while each member is working on a function, and have a schedule so we can keep on track.  The implementation phase should last a maximum of four weeks.  **4. Testing Phase:**  Our approach to testing out software is going to be a mix between static and dynamic testing. During the early stages of development we will use static testing such as walkthroughs, reviews and inspection of the code, this mean we will not execute any code during static testing. We feel the static approach early on allows us to gain feedback about how our code is progressing and if our product is of technical standard and quality for the customer. As our software progresses and we have enough feedback to move on to the next phase which will hopefully see us with much functionality coded we will begin dynamic testing of the code meaning we will execute it. We will use unit testing which will see us testing our core functionality which is not allowing a user to run into any PHP errors but redirected to a human readable error message, another way of unit testing out website will be testing the presentation of it, the user should be able to navigate freely through the site and not run into difficulties deciphering the website. We will also implement User testing for appropriate feedback on core functionality and presentation of the website. The user testing is vital to our testing as it is the ultimate way of gaining valuable feedback on how to develop, fix or maintain certain aspects of the website. |
| Deliverables  |  |  |  |  | | --- | --- | --- | --- | | Code | Name | Priority | Description | | 1 | *Working program which asks questions* | *Base* | *We first make sure that we have a working program which does what it’s supposed to do. Ask questions and allow students to answer. It tells the student what questions the user got wrong. No add ons or extras.* | | *2* | *“Rank” , “Unrank” modes , Social aspect , Code Box* | *Stretch* | *After achieving the base program, our focus is to add a rank and unranked mode. Also add a social aspect with a forum and profile page.* | | *3* | *Code Box* | *Next Version* | *Our last goal is to try and add a Code Box where the user can test the code on the fly in the program whilst doing the lessons.* | |
| Technical Requirements *PHP, MySQL, HTML5, CSS, Apache, JavaScript.* |

**Related Documents**

|  |  |  |
| --- | --- | --- |
| Document Title | Author(s) | Description |
| *Design document* | Group H | Detailed description of design for this  project |
| Assumptions and  Dependences | Fabio | A list of the assumptions and dependences need for this project |
| Event Scheduling | Ryan | A schedule for when and who is to do each activity |
| Calendar | Group H | Dates in which we will meet up and do work |
|  |  |  |

**Project Plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week 4:** | **Task**  Design Document | **Priority**  High | **Name**  Whole Team | **Description**  We must submit a document which outlines our design for the project |
| **Week 5:** | Design Presentation | High | Whole Team | Our team leader will create the Prezi for our presentation and each member will help him create our individual slides and we as a group will practice our presentation |
| **Week 6:** | * Planning Document * Planning Presentation | High | Whole Team | We must submit a document which outlines our plans for the project. Our team leader will create the Prezi for our presentation and each member will help him create our individual slides and we as a group will practice our presentation |
| **Week 7:** | Create initial prototype and work out requirements for the project  Sprint 1 | High | Whole Team | Create a bare bone prototype and a usable interface which has a question and answer system. Do research on what questions we need to ask the students. Look into books and online tutorials. Add onto the base program a Profile system where the user can login and register. |
| **Week 8:** | Update Risk Register  Sprint 2 | High | Whole Team | Update Risk Register and address them with the team, Add on Rank Mode , and Forum and try to implement Code Box |
| **Week 9:** | Test finalized product  Sprint 3 | High | Whole Team | Work on previous week’s work and review finished work and improve on them. Finish up unfinished work. |
| **Week 10:** | Deployment of Product  Sprint 4 | High | Whole Team | Deploy the finished product , obtain feedback and review |
| **Week 11:** | Exit Interviews/ Reflection | High | Whole Team | Exit interviews for the module |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk Register** | | | | |
| Risk ID | Risk Title | Probability [Hi,Med,Low] | Impact | Action |
| 1 | Delay due to asynchronous work and dependencies. | Medium | Delay | Those who have finished their work first should help the late ones. |
| 2 | Bugs on the code. | High | non-functional code | The one who programmed the code should debug it as soon as possible; he/she can also ask help to the team. |
| 3 | Lack of knowledge to do some functionality. | Low | Delay on project progress, or even impossibility to do it. | One should engage in a research to learn how to do it. |
| 4 | Lack of consistency in parts of the project, as the work will be split between the team's members. | Low | Impossibility to run all the project's parts simultaneously. | The incompatible part of the project should be adapted in order to make it consistent with the entire project. |
| 5 | Someone get sick and unable to work on the project. | Medium | Delay on parts of the project. | The work should be split between the other members. |