|  |
| --- |
|  |
|

A black text on a white background

Description automatically generated**A blue and white logo

Description automatically generated**

|  |
| --- |
| **Ravensbourne University London** |
| **Department of Computing and Business** |

**TaskQuest**

**(GTMS: Gamified Task Management System)**

**(Semester Project)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name:** | Reigis Muca | **Registration No:** | 97508723 |

|  |  |
| --- | --- |
| **Module:** | Software Design & Development |
| **Teachers Name:** | Dr. Nosheen Gul |

**Date: 08/11/2023**

**ABBREVIATIONS**

|  |  |
| --- | --- |
| **GTMS** | Gamified Task Management System |
| **TMS** | Task Management System |
| **GUI** | Graphical User Interface |

**TABLE OF CONTENTS**

[**1.** **Introduction** 5](#_Toc150332788)

[**1.1.** **Project Background and Literature** 5](#_Toc150332789)

[**1.2.** **Methodology and Software Lifecycle for This Project** 5](#_Toc150332790)

[**1.6.1.** **Rationale behind Selected Methodology** 5](#_Toc150332798)

[**2.** **Problem Definition** 6](#_Toc150332799)

[**2.1.** **Problem Statement** 6](#_Toc150332804)

[**2.2.** **Deliverables and Development Requirements** 6](#_Toc150332805)

[**3.** **Requirement Analysis** 7](#_Toc150332806)

[**3.1.** **Software Requirements** 7](#_Toc150332811)

[**3.2.** **Functional Requirements** 7](#_Toc150332812)

[**3.3.** **Non-Functional Requirements** 7](#_Toc150332813)

[**4.** **System Design** 8](#_Toc150332815)

[**4.1.** **Sequence Diagrams** 8](#_Toc150332824)

[**4.2.** **Use Case Diagrams** 8](#_Toc150332825)

[**5.** **Prototyping** 9](#_Toc150332826)

[**5.1.** **User Interface** 9](#_Toc150332828)

[**6.** **Implementation** 10](#_Toc150332829)

[**7.** **Conclusion and Future Work** 10](#_Toc150332830)

[**1.1.** **Conclusion** 10](#_Toc150332832)

[**1.2.** **Future Work** 10](#_Toc150332833)

[**8.** **References** 11](#_Toc150332834)

1. **Introduction**

TaskQuest (which is a GTMS) will be a software application designed to help users manage their tasks and boost their productivity through the integration of game elements. I have decided to take the 2 areas of interest for the idea of the project which was adding a gaming element to a task scheduling system. The system will utilise a user-friendly interface and various gamification techniques to encourage users to complete their tasks efficiently and on time. This project will involve the implementation of a task scheduling system with gamification features that motivate users to stay focused and organised. Regarding the methodology that I will be using for this project, I will be using agile since it enables for continuous improvement based on user feedback, ensuring that the final product meets the users' requirements while promoting a dynamic and efficient development process. Lastly, I made the logo for this project using Bing Image Creator.

Features that will be included:

**Task Dashboard**: Users can create, organise, and prioritise tasks through an intuitive dashboard that displays upcoming tasks, deadlines, and progress.

**Gaming Elements**: Integrate game-like features such as experience points and levels to incentivise users to complete tasks and achieve productivity milestones.

* 1. **Project Background and Literature**

TaskQuest aims to address the challenges of task organisation and productivity enhancement by integrating game elements into the task management process. This innovative approach is intended to motivate users to efficiently complete tasks through a user-friendly interface and engaging gamification techniques. Task management and productivity enhancement have become increasingly critical in contemporary work and lifestyle management, making the integration of gamification principles a promising strategy to incentivise task completion and improve overall productivity. Currently the positive impact of gamified elements on user engagement and motivation has appeared in various apps, with notable products and applications (such as Asana or [Centrical](https://centrical.com/why-centrical/)) successfully integrating gamification into TMS’ to enhance user experience and productivity.

* 1. **Methodology and Software Lifecycle for This Project**

The selected methodology for the TaskQuest project is agile since it is repetitive and adaptable. I can effectively fulfil the changing requirements by incorporating input, making appropriate improvements, and editing the TaskQuest application on a constant basis by utilising Agile principles. With this approach, the development process is more flexible and manageable, allowing me to prioritise tasks, establish realistic goals, and gradually produce a working product. By using the Agile technique, I will be able to keep my workflow organised and controlled and make sure the project moves forward quickly and within the allotted time.

1. 6. 1. **Rationale behind Selected Methodology**

As previously stated, the TaskQuest project decided to go for the agile methodology because of its built-in adaptability, flexibility, and focus on iterative development. Agile offers an appropriate structure that enables me to efficiently integrate modifications and guarantee that the finished result satisfies user expectations, especially considering the dynamic nature of the project and the requirement for ongoing feedback and adjustments.

In addition, the Agile methodology's collaborative and user-centric principles are the reason I chose to embrace it. This allows me to concentrate on developing a user-friendly and engaging TaskQuest programme that skillfully incorporates gamification components into the task management system. The Agile approach is in line with my objective of creating an interactive software solution that is high-quality, efficient and increases user productivity, as it places a priority on adaptability and user satisfaction.

1. **Problem Definition**

The primary problem that the TaskQuest project aims to solve is the ineffective task organisation and poor user incentive to finish tasks within the time frame given to them. Users of current task management systems procrastinate and become less productive because these systems frequently lack features that are engaging and motivating. In order to motivate users to finish tasks quickly, maintain organisation, and feel a sense of accomplishment (all of which will enhance their overall productivity and time management abilities) the project plans on integrating gamification aspects into the task management process.

4. 1. **Problem Statement**

The present aim is to develop a task management system that, by gamifying tasks, not only helps users organise their duties efficiently but also encourages motivation and engagement. The goal is to create an intuitive software system that encourages users to finish assignments on schedule and offers an engaging and pleasant experience all along the task management journey.

* 1. **Deliverables and Development Requirements**

A fully functional TaskQuest programme with an intuitive user interface, gamification features, and task scheduling capabilities are among the project's expected results. In order to guarantee task completion, the development requirements also require the integration of gamified features like experience points and levels. All of which should give consumers a thorough and interesting task management experience.

1. **Requirement Analysis**


5. 1. **Software Requirements**

The TaskQuest application should be compatible with major operating systems, including Windows, macOS, and Linux, to ensure widespread accessibility. In the future, it should also be designed to function seamlessly on various devices, including tablets, and mobile devices, to accommodate diverse user preferences and usage scenarios. In other words, specific and powerful hardware such as CPU’s and GPU’s are not required in order to run TaskQuest.

* 1. **Functional Requirements**
* **Task Management**: The application should allow users to create, edit, and delete tasks, as well as set task priorities and deadlines.
* **Gamification Features**: The system should incorporate game-like elements, such as experience points, and levels to motivate users and foster engagement.
* **Task Scheduling**: The application should provide users with the ability to schedule tasks and receive reminders and notifications for upcoming tasks and deadlines.
  1. **Non-Functional Requirements**
* **User Interface**: The application should have an intuitive and visually appealing interface, ensuring ease of use and an enjoyable user experience.
* **Performance**: The system should be responsive and capable of handling multiple user interactions simultaneously without significant delays or performance issues.
* **Reliability**: The system should be reliable and resilient, with minimal downtime and the ability to recover quickly from any system failures or disruptions.

1. **System Design**

7. 1. **Sequence Diagram**

**The interactions that take place when a user adds a task are shown in the sequence diagram:**

**A black background with a black square

Description automatically generated with medium confidence**

**Key:**

* External User: Triggers the process by clicking the “Add Task” button.
* User Interface: Handles the user interface interactions.
* Task Management: Manages the creation, addition, and updating of tasks.
  1. **Use Case Diagrams**

**The main interactions between the system and external users are shown in the use case diagram:**

**A screenshot of a phone

Description automatically generated**

**Key:**

* External User: Represents the users interacting with the system.
* Add Task: Use case for adding a new task.
* View Task: Use case for viewing tasks.

1. **Prototyping**

The TaskQuest project had a prototyping phase in which the task management application was developed to facilitate task creation, management, and tracking for users. The purpose of the prototype was to show off key features like adding tasks, ticking off tasks as completed, seeing pending tasks, and monitoring user progress.

User Status (Level)

Date

Add Task

Description

Task Name

Show Pending Tasks

The List Of Tasks

Mark As Completed

**As you can see from the prototype above, this is what was envisioned for TaskQuest initially.**

**Functionality Demonstrated:**

* Task Addition: Users can add tasks with details like name, description, and deadline.
* Task Completion: Ability to mark tasks as completed, gaining experience points (XP) in the process.
* User Progress: Display of user XP, level, and completed tasks count.
* Task Viewing: Viewing pending tasks in a list format.

**Conclusion:**

Prototyping was a successful way to demonstrate the core features of the TaskQuest programme. Further refinements and development could incorporate other advancements (such the usage of tooltips) and UI improvements, even though the prototype concentrated on essential task management functions.

1. **Implementation**

**6.1, Introduction:**

In this phase, the TaskQuest project was brought to life by translating design concepts into functional code. This included designing the user interface, including features like task addition and completion into practice, managing experience points and levels, and optimising user interaction in general. In order to guarantee readability, modularity, and compliance with recommended coding practices, the code was structured. The project changed from a theoretical concept to a real application that users could interact with to manage their tasks and monitor their progress through implementation.

**6.2, Software Screen:**

A screenshot of a computer

Description automatically generated

Displays the user’s level, xp and how many completed tasks the user has completed. Progress bar is also shown to visualise the xp.

Button which adds the task (once filled) into the box below.

Tooltips

Button which displays the added tasks into the box below.

A listbox which displays pending tasks which includes the name of the task, the description, and the date. Tasks can only be seen once the “Click Here To Show Pending Tasks” is pressed.

Button which marks the selected task in the box as completed which removes it from the box and adds xp.

**A computer screen with text on it

Description automatically generated6.3, Code Snippets:**

Snippet #1

A computer screen shot of a program

Description automatically generated

Snippet #2

A computer screen with text and numbers

Description automatically generated

Snippet #3

A screenshot of a computer

Description automatically generated

Snippet #4

A computer screen shot of text

Description automatically generated

Snippet #5

**6.4, Challenges Faced:**

Challenge 1:



Correction:

A black background with white text

Description automatically generated

Challenge 2:

This challenge was by far the most challenging out of all the errors TaskQuest was presenting. In essence, the “Mark As Completed” button was not functioning as intended. Firstly, I tested if the button was responsive by placing a breakpoint so that when the user clicks the button it can show me the possible errors.

private void btnMarkAsCompleted\_Click(object sender, EventArgs e)

{

// Marks the selected task as completed

if (lstPendingTasks.SelectedItem is Task selectedTask)

{

selectedTask.IsCompleted = true;

// Refreshes the task list and user status

RefreshTaskList();

RefreshUserStatus();

MessageBox.Show("Well Done For Completing This Task!", "Task Completed");

// Displays pending tasks after marking a task as completed

DisplayPendingTasks();

}

}

* So to start off with I placed the breakpoint on the same line as the code that has been circled to the left so that as soon as that code is executed I can look into solutions.
* Then this is what was displayed in the Call Stack.
* TaskQuestApp.exe!TaskQuestApp.MainForm.btnMarkAsCompleted\_Click(object sender, System.EventArgs e) Line 24 C# [External Code]
* TaskQuestApp.exe!TaskQuestApp.Program.Main() Line 19 C#

The call stack indicates that the code is indeed reaching the btnMarkAsCompleted\_Click method. After browsing Stacks Overflow and upon stumbling across [a forum](https://stackoverflow.com/questions/17615069/how-to-refresh-datasource-of-a-listbox) with a similar problem as mine, I then realised the DisplayPendingTasks method was the issue and using the forum as inspiration, I modified it to this:

A computer screen with colorful text

Description automatically generated

After modifying the DisplayPendingTasks method to directly set the list of pending tasks as the data source, it corrected the issue.

**6.5, Conclusion Of The Implementation Stage:**

To sum up, the TaskQuest application's implementation phase has effectively transformed the project from a concept into a reality. I've worked tirelessly on the code and design of a user-friendly task management system that makes it simple for users to add, track, and finish jobs. By offering a responsive and visually appealing platform, the user experience is improved by the user-friendly graphical user interface (GUI).

The application's general usability is increased with the addition of features like task completion feedback, progress tracking, and tooltips. The code snippets included in this section give an overview of the organised and carefully documented programming approach used in the development process.

The TaskQuest programme complies with the latest trends in software design in addition to fulfilling the functional requirements specified in the project specifications. Future improvements are assured of maintainability and scalability because of the tidy and modular codebase.

As we move forward, testing, optimisation, and possible iterations are made possible by the implementation phase. The attention to detail during the implementation process demonstrates the commitment to delivering a reliable and user-centric solution. Now that the application's crucial functions are in place, testing can begin to guarantee a flawless user experience and pinpoint any possible areas that need improvement.

1. **Conclusion and Future Work**
2. 1. **Conclusion**

This project provided an opportunity to apply theoretical knowledge to practical coding scenarios. I have learned the importance of careful planning, determination, and the significance of resilience in order to shape the final product. Debugging and troubleshooting skills were honed as I tackled issues and refined the app's functionality.

* 1. **Future Work**

Looking ahead, there are several ideas for future improvement and expansion:

* Integration of User Feedback: proactively seek out and take into account user feedback to fix any usability problems and improve features according to user preferences.
* Additional Features: To make the app even more functional, consider including features like task prioritisation, deadline reminders, or collaboration possibilities.
* Cross-Platform Compatibility: To make the app more accessible to a wider audience, think about modifying it for use on several platforms (mobile, tablet etc.).
* Improved Data Management: Put in place a stronger data management system that enables users to safely save and load task data.
* Personalization options: Add features that allow users to alter the application to suit their tastes.
* Login System: To safeguard user information, employ a login and registration system. Moreover, so that a particular user can save tasks and levels.

1. **References**

* panjopanjo 3, panjo *et al.* (1959) *How to refresh datasource of a listbox*, *Stack Overflow*. Available at: https://stackoverflow.com/questions/17615069/how-to-refresh-datasource-of-a-listbox (Accessed: 26 November 2023).
* MeerAcademy. (n.d.). *Button Control in Windows Application C#*. ASP.Net tutorial for beginners with examples. <https://meeraacademy.com/button-control-in-windows-application-c/> (Accessed: 26 November)
* Centrical. (n.d.-b). *Why Centrical - Centrical*. <https://centrical.com/why-centrical/> (Accessed: 26 November)