**KEELE ON THE GO (ONLINE APP)**

**Introduction**

Keele on the go web-based app is a simple proposed online system that will integrate the basic systems needed by Keele students in daily Keele life. It will link to various online student resources such the links to KLE, evision, outlook, Teams and First bus.

**Scrum tools**

Reasons for scrum tools and insight on how to make a choice.

1. Assist you in sprinting. A solid scrum tool will assist you in running sprints by allowing you to alter the backlog as needed, assign tasks, and break down epics into smaller, more manageable user stories.

2. Keep tabs on project progress and encourage retrospectives. A scrum tool should give your team (especially Scrum Masters and Product Owners) a sense of what must happen for the product to launch, even if it isn't through a burndown graphic. A significant plus is if the technology allows you to track retrospectives.

3. Encourage teamwork. Any scrum technology that makes cross-team cooperation more difficult should be avoided, because without collaboration, your team will be less likely to focus on enhancing your new product or service by incorporating input.

4. Provide transparency A solid scrum tool will show you who is working on what and how far they are along. This enables the Scrum Master, Product Owner, and team members to make appropriate changes.

Examples of some good Scrum Tools

1. monday.com

Best Scrum tool for remote teams

1. Zoho Sprints

Best Scrum board for real-time collaboration.

1. MeisterTask

Best simple-to-use Scrum tool

1. Nutcache

Best Scrum tool for managing time, expenses, and billing

1. Jira

Best for software engineering and testing

1. Targetprocess

Best for SAFe and LeSS

1. ClickUp

Best Scrum tool for customizability

1. Vivify Scrum

Best for Scrum beginners

1. Axosoft

Best for complex projects

1. Scrumwise

**Best simple Scrum software with core Scrum features**

**UML Tools**

UML tools are frequently used in computer-aided software engineering (CASE), which strives to produce and manage finished software products as efficiently as feasible. The procedure should run as error-free as feasible and produce results rapidly with the support of technical details. Mature UML tools should work effectively in an integrated development environment as a CASE tool (IDE).

**Examples**

**Gliffy**

**Advantages**

1. Integrated team communication
2. Fast operation thanks to HTML5, drag-and-drop, and a simple user interface
3. UML 2 compliant
4. Extensive support

**Disadvantages**

1. Only group businesses can access features like image export and the Google Drive interface.
2. There isn't a debugger available.
3. Creates no code

**GitMind**

**Advantages**

1. Beginner-friendly
2. The free version contains all features.
3. All popular browsers are supported
4. Templates abound.
5. Teamwork

**Disadvantages**

1. There is no version to download.
2. Creates no code
3. There is no way to connect to other services**.**

**MagicDraw**

**Advantages**

1. All diagram type templates in current UML 2.5 are included.
2. Many programming languages and file formats are supported.
3. User interface that is simple and easy to use.
4. Import/export, code generation, reverse engineering, and round-trip engineering are all examples of round-trip engineering.
5. Time to become acquainted is short.

**Disadvantages**

1. Only after registering is the file available for download.
2. Only Enterprise edition users have access to WebPortal.
3. Scrum tools used by the team and some of their inputs.

**Artifacts**

**Product backlog**

**1.** User interface and security

2. Incorporating the apps using their web address

3. Activating notifications on notification bar

4. Adding more features (language, timezones and dark mode)

**Sprint backlog**

**1.** Saturday -Gather more info about the project and UML to use

2. Sunday- Create the UML

3. Monday- Design user interface

4. Tuesday - add security

5. Wednesday -Documentation

6. Thursday - inspect and complete product to the "definition of done "

7. Friday- meetings

**Refined product backlog**

1. Product Increment that meets the "definition of done". The project was a straightforward one without complications, so the requirements and outcome didn't need any necessary refinement.

**Scrum events**

**1.** Sprint planning: The sprint planning was where we laid the foundation of the project. Pointing out the project focus, product backlog, sprint backlog, tools and method of implementation.

2. Daily scrum: This wasn't very effective as we had different timing to different activities.

3. Sprint review: This was done at the end of the implementation process is and the outcome was satisfactory for a prototype.

4. Sprint retrospective: As the team wasn't having much contact this didn't take place as it should

**Problems and proposed solutions**

Currently all students have their essential applications scattered all over their digital platforms and devices. There is no one central platform that brings together all these essential platforms together. There is the issue of some students running out of storage space due installation of several essential applications on their devices. Another major and catastrophic issue is missing out on information decimated from the university or their respective schools or faculties. This normally occurs when the information is dispatched from a different platform rather than the one the student is familiar with.

The system seeks to address these issues by bringing together these important student resources to one space. The system is web based that means that the student will be able to access most of the information from anywhere with any device connected to the internet. The notification section will ensure that the student does not miss on any important information from the platform that he/she is logged on in the system.

**SYSTEM REQUIREMENTS**

These are descriptions of the features and functionalities of the target system. They convey the expectations of the system from the software product. Requirements can be obvious, known or unknown, expected or unexpected from the client’s point of view.

**FUNCTIONAL REQUIREMENTS**

**INTERFACE REQUIREMENTS**

1. Accept username and password
2. Display the various essential school apps
3. Display the date and time
4. Display notifications

**NON-FUNCTIONAL REQUIREMENTS**

1. Security
2. Capacity
3. Compatibility
4. Reliability and availability
5. Maintainability and manageability
6. Recoverability and serviceability
7. Usability
8. Performance
9. Environment – which kind of environment will the system operate
10. Regulatory

KEELE ON THE GO

SIGN IN FORM

USERNAME

PASSWORD

ESSENTIAL APPS

KEELE APP (link)

FIRSTBUS (link)

TEAMS

TAXI/UBER

BLACKBOARD

NOTIFICATION BAR

TIMETABLE

**System Development life cycle (SDLC)**

For a software product to be functioning and working efficiently, it undergoes a series of standard activities to manage the system development through the project life span. These activities determine the success of the product. One of the common methodologies is the system development life cycle. This method is a structural approach used to plan, design, develop and test the software product quality as well s the implementation of it until deployment.

Diagram

Description automatically generated

(Figure: SDLC ((Testingexcellence 2016))

The proposed system will apply SDLC as well as SCRUM to achieve the desired software product.

1. **Planning stage**

The first stage of SDLC methodology, establishes basic structure of project, feasibility study and risk assessment. Entails listing of requirements and objectives of the project. Much information and documents on project goal are needed to achieve more output in terms of management plan structure, plan quality, overall project plan, scope and the predictable results.

Diagram

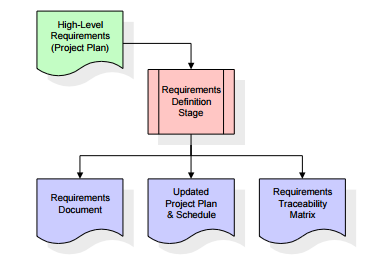
Description automatically generated

(Figure: Planning stage (Anon, 2008))

The planning phase is achieved by forming a SRUM team that will conduct SCRUM activities to realise success of the system.

1. **Analysis stage**

Identified objectives and goals are refined into requirements. Basically, in this stage all requirements are listed and how they will be realised. Requirements are clarified and documented via SRS (Software Requirements Specifications) which have all the software needs to be implemented. Also eliminates unexpected uncertainties and starts the baseline of the next phases.



(Figure: Analysis stage (Anon, 2008))

1. **Design Stage**

This is the diagrammatic representation of the SRS, either architectural or physical. These are specific designs that describe the software in detail. Once complete, each requirement will have a design element that shows how it satisfies the system needs and its interaction with the rest of the system.

Diagram

Description automatically generated

(Figure: Design stage (Anon, 2008))

1. **Implementation Stage**

Coding and programming of the project begins. This stage largely depends on the accuracy and completeness of the previous stages. The programmer/developer is responsible to producing a functional, efficient, elegant and meets requirements system. After coding the system is thoroughly tested and system information documented.

Diagram

Description automatically generated

(Figure: Implementation stage (Anon, 2008))

1. **Maintenance and support stage**

After implementation the system is deployed to the users, maintenance service follows to ensure that the system always performs to the requirements standards. Enhancements and or changes to the system are made at this stage to keep the system up to date. This stage also ensures reliable execution.

**Feasibility Study**

It is the study of productivity of a product, covers the economic, legal technological and scheduling aspects of producing the system. Gives managers a scale against which to measure practicality and decide on positive and negative consequences of the venture before investing extensive amount of time and money.

Emphasis is put on considering all substantial aspects and factors affecting the project, then decide whether to pursue the project or not. Distinguishes strategic issues and business-related issues, alongside providing answers to lighten them. The study include.

* Operational feasibility
* Technical feasibility
* Economic feasibility
* Schedule feasibility

**Operational Feasibility (PIECES Framework)**

Alludes to the measure off solving difficulties using a new framework. Operational feasibility majors on how the project fits with the current business environment and targets concerning improvement, conveyance data, corporate culture and existing processes.

To achieve desired results, operational results must be conveyed during design and development. This includes parameters such as reliability, maintainability, usability, disposability and disposability.

PIECES embodies performance, information, Economy, control, efficiency and services.

**Performance**

Used to identify response time and throughput of the system. Throughput is the amount of work done over a period, while response time is the average delay between transactional request and response to the transactional request.

**Information**

Concerns with the production of auspicious valuable and precise data the client and the client.

**Economic**

Concerns examination o project costs and income with end goal to decide if it is sensible and conceivable to pursue the project to the end.

**Control**

Recognises methods and whether they offer effective control to ensure accuracy and security of information.

**Efficiency**

Concerns with how methods of operations utilize accessible assets including individuals, time and flow of forms.

**Services**

Informs on whether current methods of operations provide reliable services. Checking the product gives dependable outcome and additionally ensuring everybody is working in association fit for utilizing the services.

**Technical feasibility**

Procedure to determine whether there are enough innovation assets to embrace the project. Currently most of the students are connected to the internet, which can be accessed anywhere anytime thanks to accessible mobile phones and computers.

**Economic feasibility**

This system will have the cost of having to purchase a domain and renting hosting space. There is also the unforeseen maintenance cost.

**Design Diagrams**

**Entity relation Diagram**

Diagram

Description automatically generated

**Context diagram**

Diagram

Description automatically generated

**Interface Design**

The following is the registration form.

Graphical user interface

Description automatically generated

**The sign in form**

Graphical user interface, application

Description automatically generated

**The keele University home page**

Graphical user interface, website

Description automatically generated

**Firstbus home page**

**A screenshot of a computer

Description automatically generated**

**Reference**

Invensis Learning Blog. 2022. Top 10 Scrum Tools in 2021. [online] Available at: <https://www.invensislearning.com/blog/top-10-scrum-tools/amp/> [Accessed 13 April 2022].

development, W. and tools, T., 2022. 6 UML tools for every occasion. [online] IONOS Digitalguide. Available at: <https://www.ionos.co.uk/digitalguide/websites/web-development/the-best-uml-tools/> [Accessed 13 April 2022].

Guru99. 2022. 25 BEST UML Tools | FREE UML Diagram Software in 2022. [online] Available at: <https://www.guru99.com/best-uml-tools.html> [Accessed 14 April 2022].

Geekbot blog. 2022. 12 Best Scrum Tools for Project Management in 2022 (Jira, etc.). [online] Available at: <https://geekbot.com/blog/scrum-tools/> [Accessed 13 May 2022].