

ft_transcendence: Project Report Requirements

This document is designed to provide students with guidelines for a structured and comprehensive project report. The report should encompass the following key sections:

- Title Page
- Abstract
- Table of Contents
- List of Figures
- List of Tables
- List of Abbreviations

The main body should consist of the following:

- Introduction
- Software Development Life Cycle (SDLC)
- Requirement Analysis
- Design
- Implementation
- Testing
- Evolution

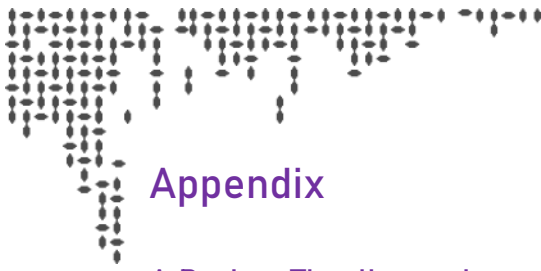
Link the following elements to the SDLC:

- Gantt Chart: Include project milestones, tasks, sub-tasks, owner, duration, dependencies, and the overall project timeline.
- Risk Register and Risk Matrix: Outline significant risks and their mitigation strategies
- Functional Requirements: Specify the software's tasks, user functionalities, interactions, constraints, and use cases.
- Technical Requirements: Specify the hardware, software, security, integration, performance, and compliance aspects necessary for the system's functionality.
- Wireframes: Include visual representations of each application screen with annotated descriptions, illustrating the layout and functionality of key elements and the sequential user navigation flow.
- Flowcharts: Illustrate the primary user interactions, system processes, and data flow across key components such as the homepage, settings, player profile, chat, and game sections, providing a comprehensive overview of the project's functional workflow.

Conclude with:

- Conclusion
- Appendices





Appendix

A Project Timeline – shown via a Gantt chart

A Gantt Chart is commonly used in project management to show project phases, activities (tasks / events) presented against a timeline.

Essentially, each activity is represented by a horizontal bar; the position and length of the bar reflects the start date, duration and end date of the activity. Creating a *bar* for each of the project activities allows a viewer to see:

- the various phases and associated activities of each phase
- the time scale of each activity (planned start date, duration and end date)
- where activities overlap with other activities, and the duration of the overlap
- the time scale of the project from inception to completion

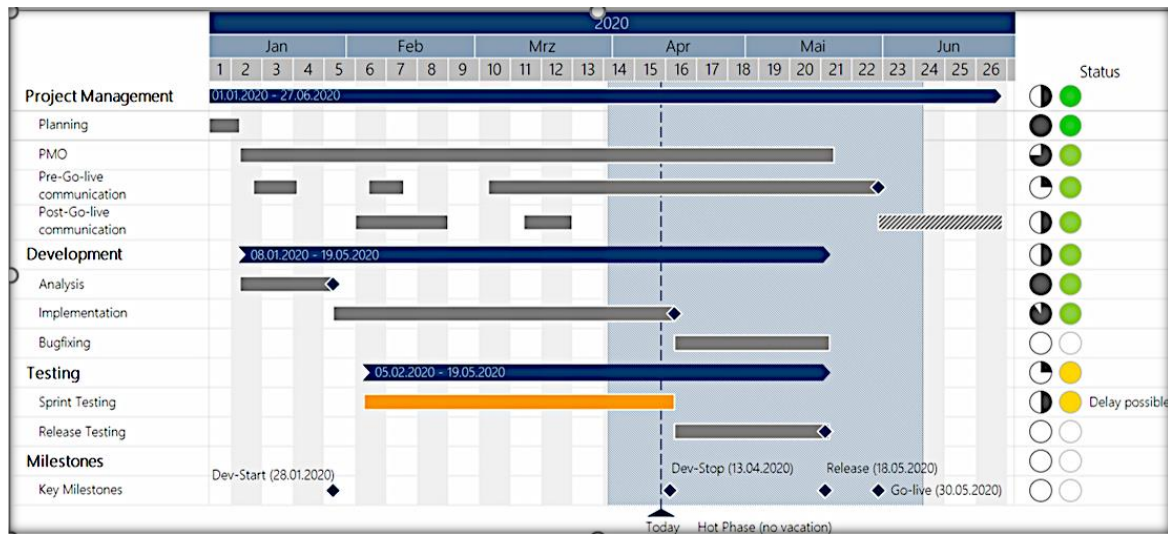


Figure 1. Example Gantt Chart





Production of a Risk Register

A Risk Register is used to identify potential project setbacks. A Risk Register aims to collectively identify, analyze, and solve risks before they become problems.

A Risk Register should include:

- a description of each possible risk
- the likelihood of the risk occurring (a numerical score)
- the impact to the project if the risk occurs (a numerical score)
- the risk owner(s)
- the mitigating action

Date Raised	Risk Description	Likelihood Of the Risk Occurring	Impact If the Risk Occurs	Severity Rating based on impact & likelihood.	Owner A person who will manage the risk	Mitigating Action Actions to mitigate the risk, e.g., reduce the likelihood.
dd/mm/yy	This is a sample text that you can edit. You can change font.	Medium	High	High	Project sponsor	This is a sample text that you can edit. You can change font.
dd/mm/yy	This is a sample text that you can edit. You can change font.	Low	High	High	Project sponsor	This is a sample text that you can edit. You can change font.
dd/mm/yy	This is a sample text that you can edit. You can change font.	Low	Medium	Medium	Project manager	This is a sample text that you can edit. You can change font.
dd/mm/yy	This is a sample text that you can edit. You can change font.	Low	Medium	Medium	Project sponsor	This is a sample text that you can edit. You can change font.

Figure 2. Example Risk Register

Finding the product of (multiplying) the Likelihood with the Impact will give you a **risk score** for each risk. The ranked risk score will provide priority areas to focus on.

Figure 3, a Risk Matrix, shows the product of Likelihood and Impact and some 'typical' risk score descriptors.

To note: it is possible that certain projects may wish to deploy certain weightings to particular risks irrespective of the product to ensure the risk is suitably monitored.

		Impact How severe would the outcomes be if the risk occurred?				
		Insignificant 1	Minor 2	Significant 3	Major 4	Severe 5
Probability What is the probability the risk will happen?	5 Almost Certain	Medium 5	High 10	Very high 15	Extreme 20	Extreme 25
	4 Likely	Medium 4	Medium 8	High 12	Very high 16	Extreme 20
	3 Moderate	Low 3	Medium 6	Medium 9	High 12	Very high 15
	2 Unlikely	Very low 2	Low 4	Medium 6	Medium 8	High 10
	1 Rare	Very low 1	Very low 2	Low 3	Medium 4	Medium 5

Figure 3. Risk Matrix





Functional and Technical Requirements

Develop a list of the major functional and technical requirements for the game. Be sure to refer to the Project Descriptor for the general requirements of the game.

Functional Requirements

A functional requirements document contains the details of product features or functions those developers must implement to enable end users to accomplish their tasks.

Examples include:

- The system must allow the user to enter data
- The system must generate reports every 24 hours
- The system must send a confirmation email whenever an order is placed
- The system must allow users to verify their accounts using their phone number

Technical Requirements

Technical requirements include the details of how the functionality of an application can be achieved.

Examples include:

- Performance: The system will have an average page load time of less than 2 seconds.
- Serviceability: Changes and upgrades to the system will not require total outages.
- Privacy: The user interface will not allow employees to view customer birth dates stored in the customer database.
- Information Security: User credentials and all personal identifiable information will be encrypted in storage and transit.
- Data: User data will be sourced from the customer database without permanently storing it in the system.

In simple terms, functional requirements are about what you want from your software / application, and technical requirements are about how you get there.





Wireframes

Wireframes are visual guides in which designers propose elements for screens and webpages and show how experimental solutions would flow for users.

Wire-framing is invaluable early in the 'interaction design process' for design teams to explore how concepts accommodate user needs. Good wireframes are skeletons for powerful prototypes and delightful designs.

In your Wireframes you should show how your game considers Human Computer Interaction (HCI) / User Experience (UX)/ User Interface (UI).

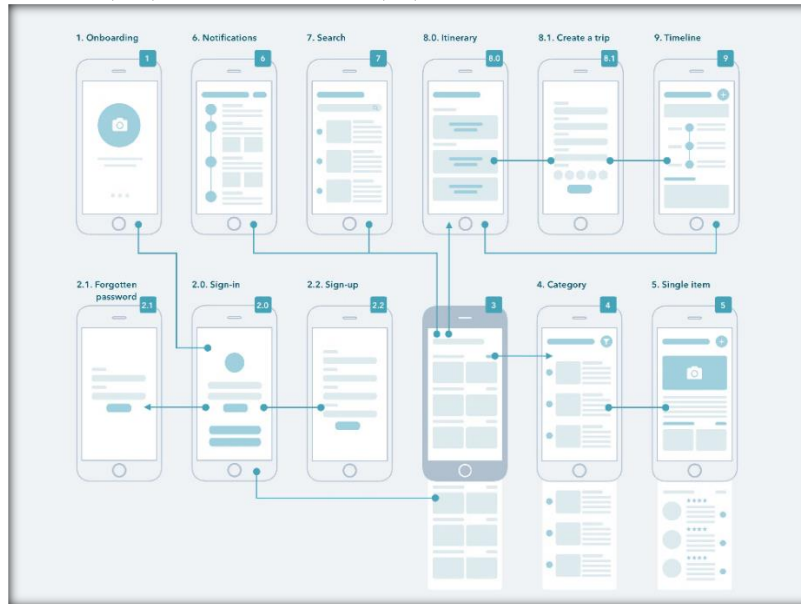
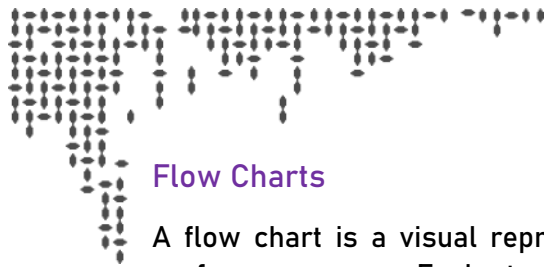


Figure 4. Wireframe of a Traveling App





Flow Charts

A flow chart is a visual representation of the sequence of steps and decisions required to perform a process. Each step in the sequence is represented by a shape. Steps are linked by connecting lines and directional arrows.

A flow chart is a powerful tool that, when properly designed and constructed, communicates the steps in a process. This allows anyone to follow the process logically from start to end.

To note: There is a standard protocol for the shapes used in flow charts. See for example <https://www.lucidchart.com/pages/flowchart-symbols-meaning-explained>

Some examples of Flow Charts are shown in figures 5-7

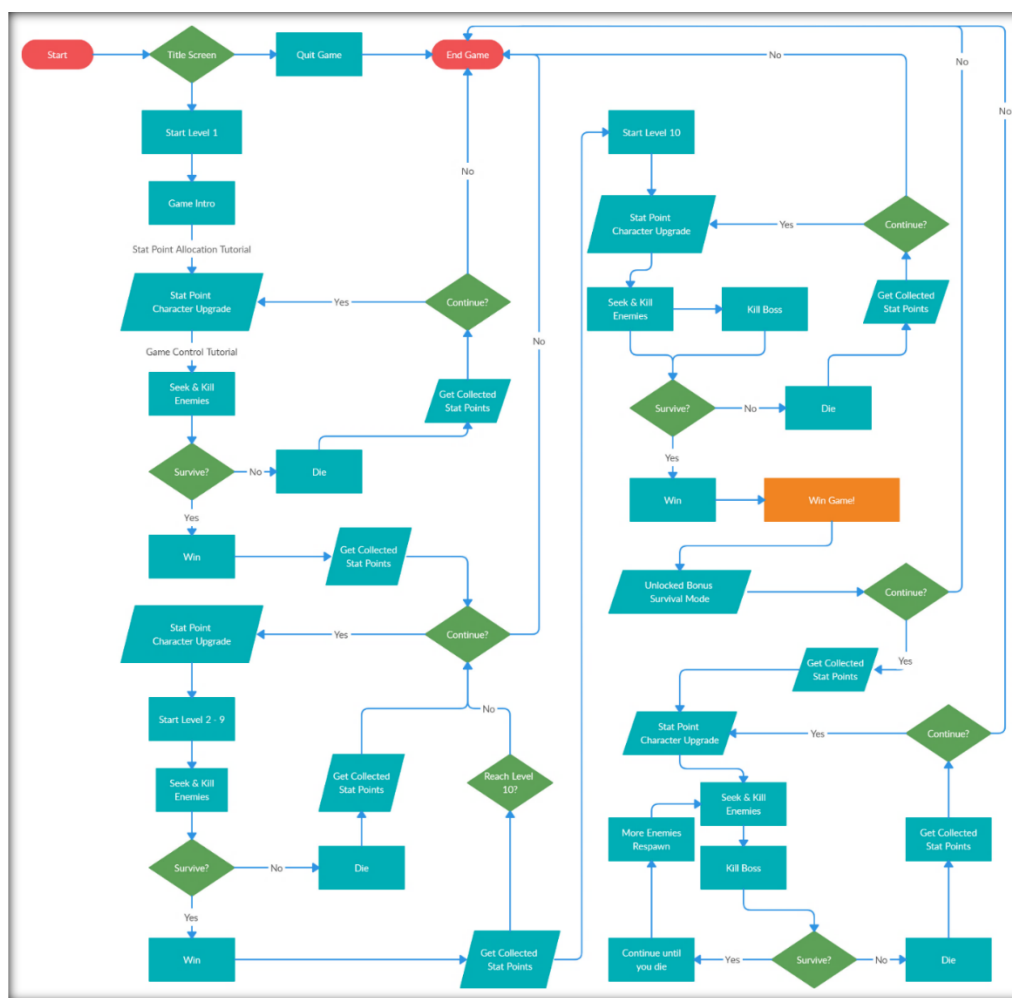


Figure 5. Action Role Playing Game Flow Chart



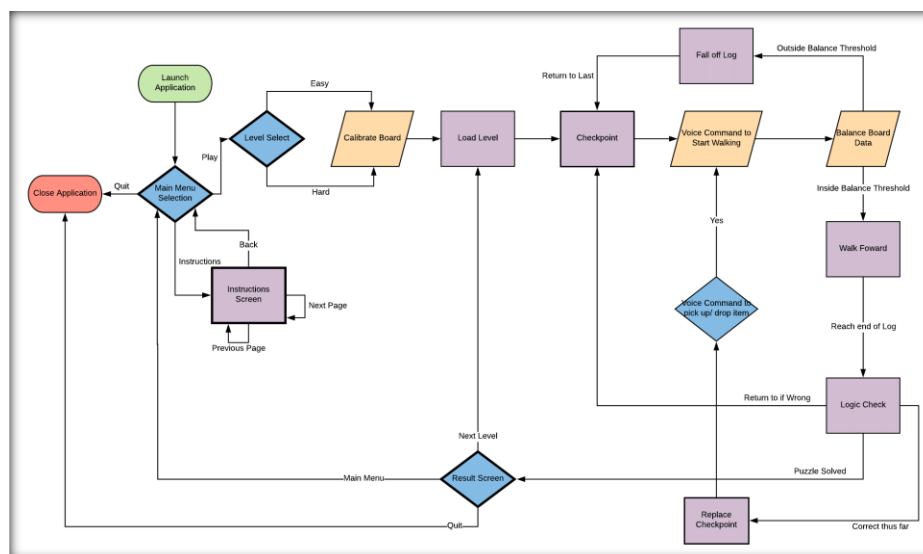


Figure 6. 'Don't Fall: The Game' Flow Chart

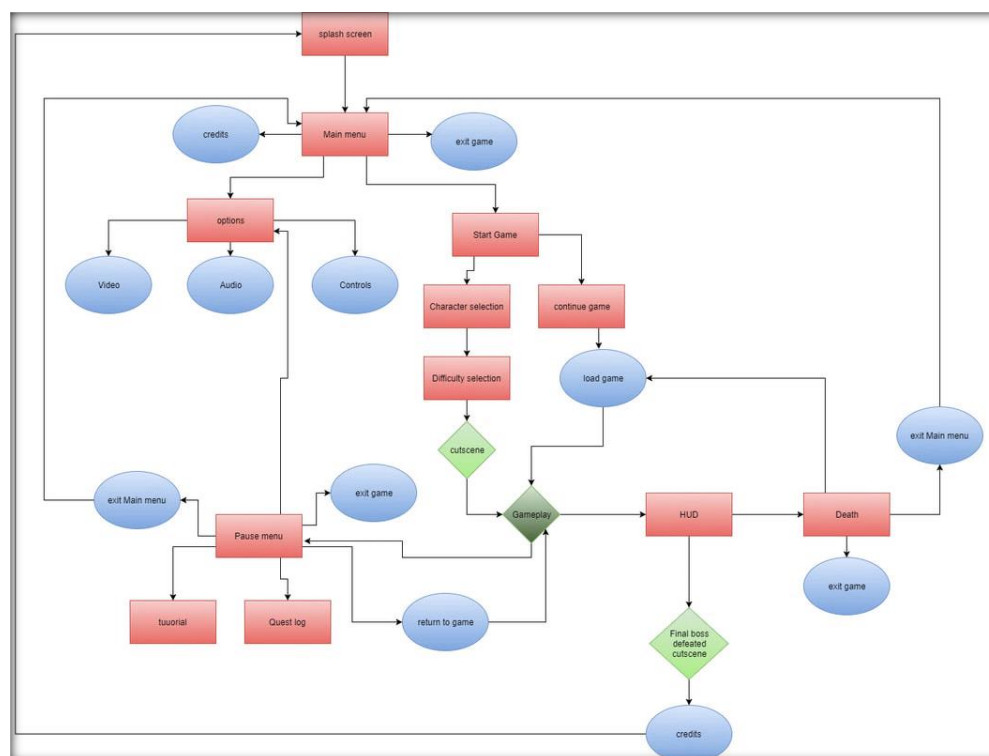


Figure 7. 'Alan Horton's Game Design' Flow Chart

