Project Handbook

X02 EMPLOYABILITY ASSESSMENT TOOLKIT

This template is loosely based on the following standards:

- Australian Standard AS4071-1992(R2014) - Software project management plans
- AS/NZS ISO/IEC/IEEE 42010:2013 – Systems and software engineering – Architecture description

Note that following this template is not enough to claim conformance to either of the above standards! For Project courses, some sections have been excluded completely, and some are optional. These are noted, and may be skipped

FEDERATION UNIVERSITY

Muhammad Daniyal (30385522)
TEAM X02

!IMPORTANT:

"In this part, I have discussed about vision statement section only as per assessment requirement."

Revision

Use whichever style of versioning you prefer.

You may also include the main authors of each change, and the list of pages that have been changed

Version Number	Date approved	Approved by	Description
1.0	2016-01-01	Team Member	Initial release of plan

Preface

Describe the purpose and audience of this document, in your own words.

Table of Contents

List of Figures

No table of figures entries found.

List of Tables

No table of figures entries found.

Vision Statement

To commence with, the rudimentary goal of this project is to develop a comprehensive employability assessment portal which will be comprised of three main functions i.e., a career assessment tool, a Curriculum Vitae (CV) assessment tool and a career development planning tool. It will benefit the stakeholders in many ways by saving time, managing large volume of candidates, providing more efficient review process and sorting resumes based on keyword identifiers provided by the employer. Basically, CV screening determines whether a candidate is prepared and qualified enough to be considered for a job role based on education and experience of the candidate. It will assist and advice students and institutions on workplace trends and employability preparation. The career assessment tool will help students to identify their target job roles by assessing their personality types, preparation level and fitness score for the preferred employment. The CV assessment tool will allow students to upload a CV and subsequently, provide them with a detailed feedback based on preferred job roles. The career development planning tool will recommend students with a list of solutions for competency development. Overall, it will allow students to map their current competencies with target audience by assessing **https://employability.life/** assessment portal.

1. Introduction

1.1 Project Overview

*Give a **summary** of the project objectives and deliverables, and any other work that required as part of the project.*

Include a brief description of the resources required, deadlines, and budget.

1.2 Project Deliverables

Describe what items are to be delivered to the client, approximate dates, and quantities (if any).

You do not need to include process documentation (such as sprint documentation, design documents, or similar) here, but should include a user manual, an installation manual, and technical documentation.

1.3 Evolution of the Handbook

Plan for making scheduled and unscheduled updates to this handbook. How will you keep it up to date?

Consider:

- *When will scheduled updates happen?*
- *Who is responsible for updates?*
- *How will you put the handbook in change control?*
- *How will everybody be notified of handbook changes?*

1.4 Reference Materials

This is a complete list of materials referenced elsewhere in the handbook, such as style guides, coding standards, documentation standards, methodologies, etc.

Indicate if you haven't used any external references.

Use any style that you like. If you don't know any good ones, then use IEEE or APA style

http://www.ieee.org/documents/ieeecitationref.pdf

http://www.apastyle.org/

1.5 Definitions and Acronyms

Define, or provide references to the definition of, terms, acronyms, or abbreviations used in the handbook.

Term	**Definition**

2. Organization

2.1 Process Model

This section should describe how the project functions and activities (i.e., the work you are doing) work together to build your project.

*You should include a high-level breakdown of the activities, with a rough timeline. Include a chart, diagram, or timetable. You should indicate this at the level of **each sprint**. For each sprint, clearly state the outcomes and deliverables to be produced.*

Include preliminary agreed dates for sprint review meetings for demonstrations to your client.

As this is an agile project, this is necessarily a projection/estimate rather than a binding timeline.

2.2 Organizational Structure

*Describe the structure of the project team, from a **process perspective**. Identify scrum roles, and how you will determine changes in these roles.*

2.3 Organization Boundaries and Interfaces

Describe the "administrative and managerial boundaries" between you and your client, and other stakeholders or contributors.

Be specific – indicate people and their roles – the more specific you can be the more useful you will find this document.

How will client communication be handled, who will be responsible, how often will you be in contact?

2.4 Project Responsibilities

*Describe the **non-procedural** roles of each of the team members – for example who is responsible for design, programming, artwork, quality assurance and testing, user documentation, technical documentation, etc.*

You can use a matrix if the team members share responsibility for each function (which is recommended).

3. Managerial Process

3.1 Management Objectives and Priorities

What is the management philosophy? Are you aiming for high performance, high equity, flexibility, or learning new skills? Sometimes you will need to choose, so how?

This is also good place to address conflict resolution, consider how you will handle interpersonal problems and how you will resolve them.

3.2 Assumptions, Dependencies, and Constraints

State:

- *The assumptions upon which this project is based*
- *The external events or inputs that the project depends on*
- *The constraints under which the project is operating, for example budgetary, staffing, availability, hardware.*

4. Technical Process

4.1 Methods, Tools, and Techniques

Detail the tools and techniques used to build the project – note that this isn't necessarily limited to the target platform, but includes your project management, documentation, and communication tools.

Describe your team's implementation of the Scrum framework. If you like, you may refer the reader to external documents.

*What tools will you use to handle **communication** within your team? (e.g., MS Teams)*

*How will you specify and model your **software designs**?*

*Which **document and code management** systems are you using? (e.g., MS Teams, GitHub)*

4.2 Software Documentation

What is the plan for creating user and technical documentation?

You will need to plan for the creation of a User Manual and an Installation Manual

How will documentation be reviewed and tested for accuracy?

Will you use a style guide? If you use an external guide, be sure to include it in your references.

5. High level Project Plan

Identify how many sprints are planned for the project. For each sprint, identify the sprint goal – the key objectives that you forecast achieving in that sprint. For each one, ***indicate how you will evaluate it as successfully 'done'*** *(this is a bit like a high-level condition of satisfaction). "The sprint goal is an objective that will be met within the Sprint through the implementation of the Product Backlog, and it provides guidance to the Development Team on why it is building the increment." (The Scrum Guide, p10)*

You may decide that in the first sprint, you are doing some research and building a prototype to demonstrate what is possible with some particular technology, or you may plan that you will build a particular component of your product that is key and high priority. In true agile, in each sprint, you should be developing an increment or release of your product or some deliverables relating to your project. In the context of the learning experience and as you are students without significant experience, you may plan to focus on some early design decisions in your first sprint, but ***please ensure you identify clear goals and outcomes for each sprint.*** *So, it may be that you are designing a database, but don't stop at an E-R diagram, design the database and then create an implementation of the database and some queries that are useful for achieving key objectives relating to your product. Alternatively, it may be that you are designing a level for your game, so identify (some of) the characters, (some of) the scripts and behaviours and then implement (part of) that level.*

At the beginning of each sprint, you will revisit this plan and then build your product backlog items to identify specific tasks to be completed during the sprint. During each sprint, you should be engaging in design, development, and testing.

"Having set the Sprint Goal and selected the Product Backlog items for the Sprint, the Development Team decides how it will build this functionality into a "Done" product Increment during the Sprint. The Product Backlog items selected for this Sprint plus the plan for delivering them is called the Sprint Backlog… As the Development Team works, it keeps the Sprint Goal in mind. To satisfy the Sprint Goal, it implements functionality and technology. If the work turns out to be different than the Development Team expected, they collaborate with the Product Owner to negotiate the scope of Sprint Backlog within the Sprint." (The Scrum Guide, p11)

Provide a link referencing your online project board (e.g., in GitHub project board or Trello board) here.

6. Non-functional Requirements

For each of the following section headings, identify any relevant non-functional requirements. For each one, ***indicate importance*** *and* ***how you will evaluate it*** *(this is a bit like the conditions of satisfaction and makes sure you can measure your success)*

Don't include spurious requirements just for the sake of it!

If your project has no relevant non-functional requirements for any of the following domains, leave the section heading in-place and indicate that there are no applicable requirements.

6.1 Platform

Platform requirements relate to the hardware and software environments that your system must operate within

6.2 Communication

Your project may need to interact with other systems. Only include non-functional requirements here – ***how*** *your system communicates, but not what it communicates or why. Protocols, frequency, message latency, maximum message sizes, flooding, or authentication considerations might belong here. Note that these requirements should be limited to communication with other systems – not with the user.*

6.3 Performance

Indicate relevant performance requirements. Consider frame rate, response time or time budgets, input latency, network utilisation, CPU use, battery use.

6.4 Security and Privacy

Indicate any security and privacy requirements. Consider the security requirements around user authentication, what information you should/should not store, encryption, password storage, backups, what should/should not be included in log files or error reporting

6.5 Audience, Usability and Accessibility

Who is using your product? What requirements arise because of this audience? Consider language, internationalization/localization, pre-existing knowledge, familiarity with other tools. Usability and Accessibility are related to audience.

6.6 Reliability

Consider requirements around system availability, up/downtime, fault logging, redundancy, error tolerance, etc.

6.7 Modifiability

If your system must be modified or updated, how does this need to happen?

6.8 Economic

There are likely to be economic constraints/requirements on your project development. Indicate these requirements here.

6.9 Legal

Applicable regulatory or legal requirements. Consider also licensing, certification, etc.

6.10 Standards

In some cases, you will need to adhere to existing standards for file formats, network systems, or to be compatible with other systems or products. These requirements may apply to the project or the development process.

6.XX Other Non-Functional Requirements

Include any other non-functional requirements you identify here. Give each additional section an appropriate number and title.

7. Software and Systems Architecture

7.1 Architecture objectives

Describe the desired properties and goals of your system architecture. You may refer to the above non-functional requirements where necessary. This section should be only a paragraph or so.

7.2 High-level architecture

*Describe the overarching design of the system, or at least your current plans for the architecture. Examples of your software and systems architecture might be **n-Tier**, **distributed**, **microservice**, **monolithic, Model-View-Controller, Model-View-View-Model** or a combination of several of those.*

The following sections are un-numbered as you may not need to include some, depending on your project. You should number sections appropriately

7.X System context

Where does your system fit in with other systems? How and why does it interface with them? How is responsibility for functionality split across systems?

7.X User Interface / Interaction Design

Include initial user flows, visual designs, mock-ups, concepts, sitemaps, or any other appropriate documentation to show how you anticipate users will interact with your system

7.X Data model and software design

Describe your initial database design, using diagrams or data dictionaries. Indicate if you are using any standard data design patterns or conventions.

If you are designing a file format or new data structures, describe the format, your justification for its design, and similar formats.

You may include other types of system design diagrams here too; choose whichever diagrams best suit both your project and your team's design process.

7.X Assumptions

You may make certain assumptions about your target platform/system when creating your design. Indicate those assumptions here.

Examples might include number of users, frequency of use, software libraries, available bandwidth, database size, hardware revisions (i.e., which phones does your app work on?)

7.X External Dependencies

These are external dependencies in the architectures – for example are you relying on third-party systems to remain available? Library or operating system code which you can install permanently is not a dependency for this section.

7.X Concept art, storyboards

For game and multimedia projects, include appropriate concept art, character designs, treatments, storyboards, etc.

Additional Components

Include any other components here that you think are necessary, such as training plans, data conversion plans, maintenance plans, etc. Number each new section as above, starting at section 7

Index

An index is optional. If you choose to include one, explore whether your word processor can do so semi-automatically for you.

Appendices

Any supplemental items (such as change request forms, etc.) that do not form part of the handbook proper should be included as appendices.