Assignment 1 – Project Diary

Team Members

* Name, Surname – Student ID
* Name, Surname – Student ID

# Project Goal

Describe here what is the aim of your project (a couple of paragraphs are enough). They should highlight clearly what the project does and its core function / feature.

# Project Value

## Value Hypothesis

Describe here what will make this project successful and why it is meant to be successful.

Example:

“move my business model from physical shop to an e-commerce site to increase reach and sales revenue.”

## Growth Hypothesis

How this project will continue to live and grow?

Example (with reference to previous):

“Increase my customer base by targeting specific facebook groups, and plan for horizontal scaling (increase the number of web servers) as the customer base grows”

## Metrics

How will you test/prove whether you are successful?

Example (with reference to previous):

“We will assess our success by looking at the sales performance in the next quarter and by instrumenting our online website to monitor the user behaviour and understand whether there is a increase of interest in our business.”

# Project Plan

## Communication Strategy

Describe here how do you organise project internal communication (e.g. emails, meetings, or other forms of information exchange).

## Task Management Strategy

Describe here how do you divide the work among the project team members and how will you keep track of the project lifecycle.

## Iteration Plans

Describe here how many iterations you plan to use to develop the project and what is the expected result at the end of the iteration. Copy the tables below for as many iterations you plan to do.

|  |  |  |  |
| --- | --- | --- | --- |
| **Iteration 1** | Put Title here | | |
| Start Date |  | End Date |  |
| Description |  | | |
| Result | What did you achieved, learned? What happened? | | |
| Follow Up | Put what will be your next step (Pivot / Persevere). | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Iteration 2** | Put Title here | | |
| Start Date |  | End Date |  |
| Description |  | | |
| Result | What did you achieved, learned? What happened? | | |
| Next Step | Put what will be your next step (Pivot / Persevere). | | |

# Design Thinking Process

Refer to the [“Design Thinking Questionnaire”](https://d2l.deakin.edu.au/d2l/le/content/392209/viewContent/2851148/View) explained and used in Practice 4. Include a completed version of the questionnaire as attachment to the project diary.

# Architecture

## Overview

Put a diagram here containing describing the architecture of the project and explain in a couple of paragraphs what are the key components of the solution and what they do.

## Architectural Decisions

List here the key decisions that you have faced from an architectural perspective. Use the template provided as an example to list all the decisions that you think have been relevant in shaping your design and architecture.

Example template:

|  |  |
| --- | --- |
| **AD001** | (Title) **Choice of the storage technology and model** |
| **Problem Statement** | (What is the problem being addressed?)  Our application, even though not data-intensive, requires storage for persisting key information that is required for its function. Different storage technologies do provide different approaches to storage and impose different constraints on what can be saved. The choice of the particular storage technology and model will impact the design and implementation of other components of the application. |
| **Available Options** | |
| **Option 1.** | **Utilise a Relational Database**  Description:  This solution implies the use of relational data store based on SQL. All the entities of our system will be persisted as records in one or more tables. The solution will have a defined schema for the data model.  Pros:   * Powerful query language that can be used to operate on the data. * Model very simple to understand and known very well by the team. * Record validation is performed by the database. * Ease of availability of product implementing this model.   Cons:   * Customisation of the data model is hard, because the model is optimised for data with regular structure (i.e. records). |
| **Option 2.** | **Utilise a NoSQL / Document Oriented Database**  Description:  This solution implies the use of a document-oriented database (NoSQL) such as MongoDB or Cloudant/CouchDB. The entities will be persisted as documents in the database and we will be able to persist different (also user defined) entities within the same containers because no schema is defined.  Pros:   * Schema-less implementation. * Ease of availability of product implementing this model. * Highly scalable solution.   Cons:   * Record validation needs to be implemented within the application or by using additional libraries. * The capabilities of the query languages differ from product to product as there is no standard in the field. * Lack of knowledge of this type of storage model within the team. |
| **Option 3.** | **Utilise a File based System**  Description:  This solution implies the use of files to persist entities within our application.  Pros:   * The model is easy to customise as we can decide what to store in the file and these can either be records of the same structure or different records.   Cons:   * The set of built-in services and capabilities strongly varies from product to product, ranging from simple file access to more sophisticated operations. It might be hard to have available a query language that can be effectively used. * The model might not be able to provide backup and custom implementation is needed for this feature. * Performance might be another issue on top of custom development for accessing and manipulating the entities in the storage. |
| **Decision** | |
| **Selected Option** | The selected option is **Option 1.** |
| **Justification** | (Explain why you choose that option, essentially what based your decision on).  Option 1 is particular advantageous because of the nature of the application we’re developing. In particular, we do not need to provide user-defined records and the application entities abide to a well-defined structure that is more effectively represented and manipulated trough a relational model.  Moreover, the team has a well-developed set of skills and expertise with relational database and this will boost the development activities and reduce time. |
| **Implications** | (Describe the impact of selecting the specific option mentioned above)  We will need to provision a database solution in the cloud as a service. We might be limited in the choices of available product especially if we want to maintain the solution within one single platform of the cloud computing vendor.  We will need to find client libraries that enable us to talk to the specific database implementation. |

**NOTE:** while developing a project you are implicitly taking these decisions and go through a process that covers the items identified in the template table. The Architectural Decisions artefact (i.e. the collection of tables as the above one, one for each decision) is simply a way for improving the accountability of your actions and reflect more attentively (e.g. by putting pros and cons of the different options on paper) on the choices you make, learn from it and keep it for the future.

# Retrospective

Here you are asked to share some thought about the overall process. This should be a critical analysis of how you (as a team) performed to achieve the desired outcome. In particular, some of the elements that may guide you in this discussion are:

* How you broke down your development process into iterations.
* The learning at the end of each iteration.
* What went wrong and what went right, what would you have done differently?
* Was the original idea sound enough to pursue development successful?