Assignment 1 – Project Diary part A

# Additional Information

This is an example, that should show you the kind of information should be put in evidence. Your work will be required to be more elaborated, hence only take this a real life example.

In part B, I will complete my architecture and my retrospective.

Team Members

* Alessio Bonti

# Project Goal

As an avid Magic The Gathering player, I have to face one big issue every day, finding cards to play, some of these cards may become extremely expensive, some of them instead , are cheap, but hard to find, because people may have them but don’t’ keep track of them. Now days the ways to buy/trade cards is by

* Joining local facebook groups
* Buy from the store
* Randomly meet people in the game stores and trade

I intend to build a mobile application, which will facilitate my trading life. It will work similarly to Ebay and also Airtasker, as in it will be a selling platform, and also trading (I make a card available, others who are looking for that card will be prompted).

# Project Value

## Value Hypothesis

I will improve the MTG trading locally, this will make my life easier, it will give me ‘recognition in the community’ and could also lead to an extensive business plan, as it has the possibility to disrupt the local game stores industry.

## Growth Hypothesis

I will contact the community most influential traders and use them as my pivot to launch the application, of course I will also use the community to publicize it and will try to get the shops on board in a “join me or fight me”

Eventually, it will have to scale both horizontally and vertically, which will not be a problem. There are examples online and case studies, for example code fight, which describe something done very similar, it was built on meteor for fun, and then it found itself at a point where it had to scale up. Hence I can use this example for my technology growth.

https://www.meteor.com/case-studies/codefights

## Metrics

I will monitor the application usage, and the exchange rate as well. I am expecting a slow start, but because the application can run on a very low budget, it can move on on a ‘amateur’ level until it gets critical mass.

# Project Plan

## Communication Strategy

Since I am a one-man team, I will not need one, but because it is very standard for me, I will setup a slack channel for this project, with a bot connected to my git repo.

## Task Management Strategy

As my task management is relatively simple and very linear, I will use Trello to take care of my tasks.

## Iteration Plans

See attachment. I have used a more advanced iteration plan table, please use the ones provided with the original document.

MTG trader - iterations 1-3.pdf

# Design Thinking Process

I have skipped this part simply because I interviewed myself, you can follow my example that was uploaded in the pracs.

[**Practical 7 -design thinking alessio**](https://d2l.deakin.edu.au/d2l/le/content/392209/viewContent/2851149/View)

# Architecture

## Overview

## 

The core application takes care of the user management and all of the basic interactions, mongolab holds the data and is handled through the ddp by the meteor server. Twilio will be an extra addition, if I don’t wrap my application, I will not be able to use push notifications, hence I will instead use sms to speed up things.

The second part (top most) has to do with the ability to post items on twitter and monitoring the application performance.

## 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?