

+JMJ+

Software Design Description for Oxword Fictionary

Version 1.0

Prepared by Ben Campbell

The Fictionary Team

March 12, 2021

Table of Contents

Table of Contents.....	2
1 Introduction.....	3
Scope	3
Overview	3
Definitions	3
Considerations for Producing a SDD	3
2 Design Description Information.....	4
Introduction	4
Architectural Goals and Constraints.....	5
Architectural Views.....	6
Goals and Plans	7
3 Design Description Organization	8
Introduction	8
Decomposition Description	8
Dependency Description	9
Interface Description	10
Detailed Design Description.....	11

Revision History

Name	Date	Reason For Changes	Version
	3/12/2021	First version	1.0

1 Introduction

Scope

This document is the Software Design Description (SDD) of Oxword Fictionary. It describes the goals, constraints, plans and implementation of the app, and the process of its development, distribution, and maintenance. It's purpose is to provide a broad overview of everything, and although it does contain detail in some areas, it is not meant to be exhaustive.

Overview

This SDD is contains two parts:

[Design Description Information](#) – An overview of the project from several perspectives, and a description of the quality and business goals, as well as the issues and constraints we have and our plans to overcome them.

[Design Description Organization](#) – A detailed description of the structure and functions of the app, including the business functions, user interface, and the logical structure of the software.

Definitions

SDD: Software Design Description; A document that describes a plan for how a project works by looking at it from several perspectives. This is an SDD.

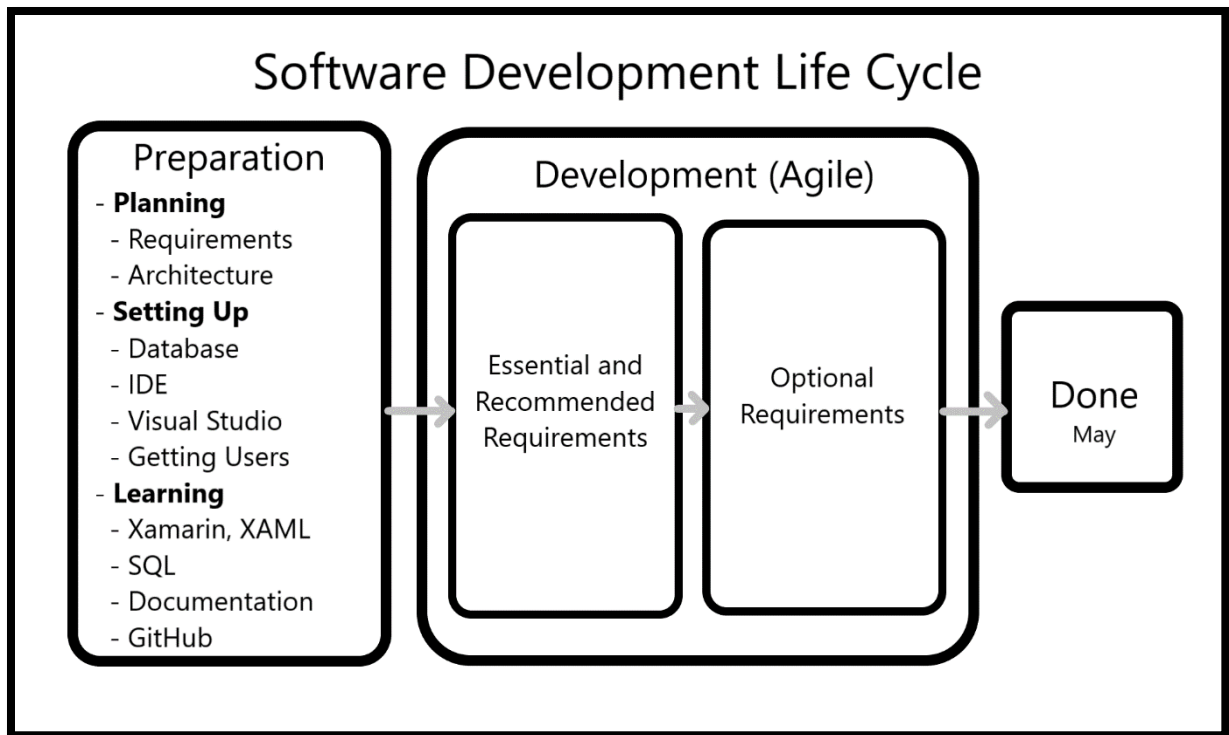
Oxword Fictionary: The fake dictionary app that we're making.

View: A view of the project from a specific type of person's perspective, such as a user's view, a programmer's view, or a businessman's view.

Considerations for Producing a SDD

Software Life Cycle

Here is a little diagram of the software development life cycle of our app:



SDD Within the Life Cycle: This SDD is part of the Preparation phase, which is the first phase in the life cycle.

Purpose of an SDD

The SDD provides a detailed plan for the important decisions and structure of the project in all its aspects. It can be thought of as the blueprint for the project, and allows rest of the development process to proceed in a structured fashion.

One important way it does this is by looking at the project from different *views*. A view is an examination and description of the project and its parts from the perspective of a specific type of person such as the end-user view, the developer view, the marketing view, or others. The different views used in this document can be seen in the section below, [Architectural Views](#).

2 Design Description Information

Introduction

The Design Description Information is an analysis and description of the project, our goals, constraints, the various issues we may come upon, and how we plan to overcome these issues. It has three sections:

1. Architectural Goals and Constraints: The goals for our project, and the things that might constrain us that we need to be aware of.

2. Architectural Views: An examination of the project from several perspectives: that of the users, logic, and deployment.

3. Goals and Plans: A more detailed description of our goals and what we plan to do if something goes wrong.

Architectural Goals and Constraints

- **Technical Platform:** People can download our app from GitHub.
- **Transaction:** n/a
- **Security:** We don't have many specific security goals, as long as our app doesn't break people's phones or open up vulnerabilities. Not much personal data will be used, only the account information for this specific app, which if stolen, will allow the thief to post comments and definitions as the person whose account they stole. Since these will be moderated by admins anyways, this doesn't seem to be a super big problem.
- **Persistence:** The data will be kept persistent through accessing it from our database.
- **Reliability/Availability:** Not much availability is needed, just enough for the developers and few users to access the app when needed, as well as for our teacher to grade it.
- **Performance:** At least five people will be able to use our app at the same time without noticing any problems.

Here are basic response times needed to keep users engaged:

- *Instantaneous reaction:* 0.1 second. This is how long it should take for a button press to react, for example.
 - *Regular flow of actions:* 0.1 to 1 second. Try to keep page transitions within this time frame.
 - *Loss of attention:* over 10 seconds. If people have to wait this long they'll start thinking of other things and getting bored, so if they're expected to wait, have something else for them to do in the meantime (at least an estimate for how much longer they'll have to keep waiting!)
- **Internationalization:** Only in America, in English.

Architectural Views

There are three different ways to look at the project and the things that it does. This specific SDD has three views:

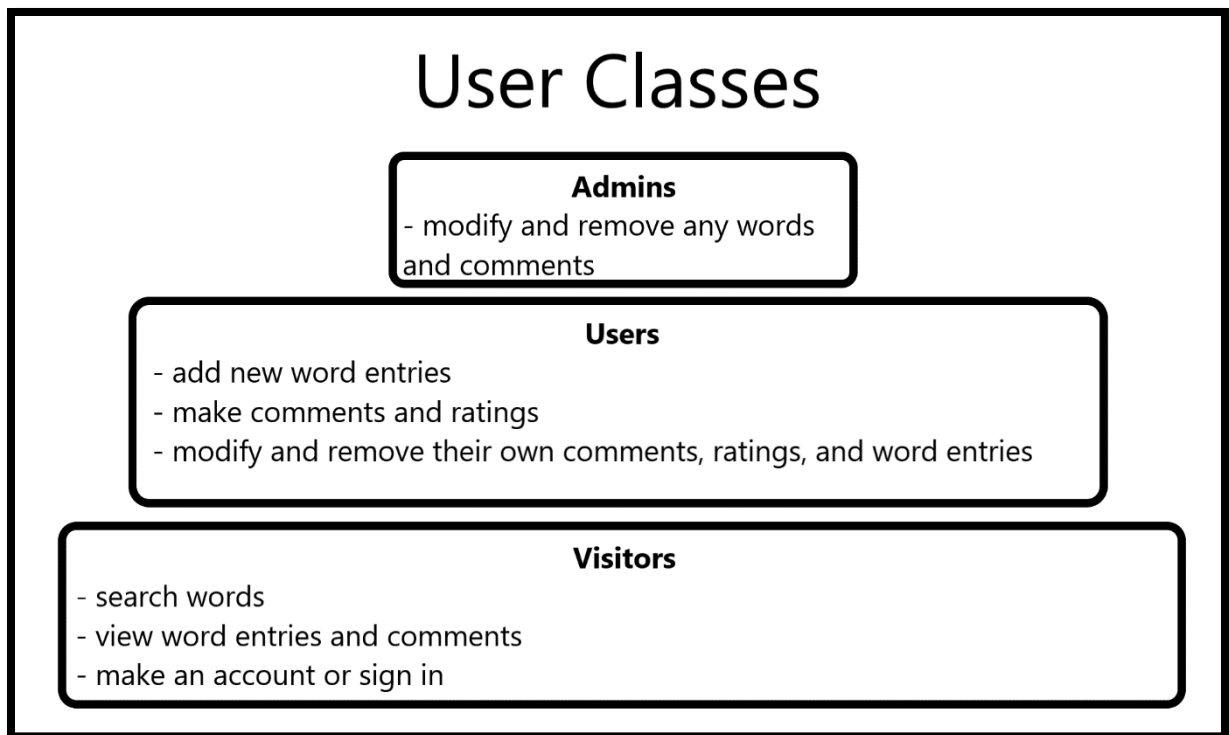
Use case view: For users. It contains the different user classes and the things they will be able to do.

Logical view: For managers and developers. It shows the different logical parts of the project.

Deployment view: For managers and developers. It shows how the data will flow when the app is deployed.

1. Use Case View

This is a description of the different types of users and the things they can do. It's best explained with a diagram:



There are three tiers of users: admins, users, and visitors. The higher tiers can do everything the lower ones can do, and more.

2. Logical View

This is a view of the basic logical sections of the entire project, including the app itself, as well as the things we use to develop it and the places where information is processed and stored. This project has four logical tiers:

1. *Client Tier*: The phone application where the users make and view words along with all the other app functionality.
2. *Web Tier*: All the online stuff that users have access to. This will all be on GitHub. It will include our source code, wiki, and other documentation for both users and developers.
3. *Data Tier*: We will have a database hosted by DreamHost that will store all our words, definitions, user info and passwords, etc.
4. *Business Tier*: This is all the behind-the-scenes stuff we use to develop the app. Since we're students, this is really just our personal laptops where we install and use all the programs we need like Visual Studio.

3. Deployment View

This is a description of how the information will flow between the various parts when the app is being used and developed.

Using the App: The app will run on users' phones and send and access information from our database through the internet.

Development and Updating: We will develop the app on our laptops and access our database and GitHub using the internet. When a new release comes out, we will upload the installable file to GitHub, and users will be able to download it and install it on their phones.

Goals and Plans

Size and Performance

We have no goals for how many users to get. See the section above, [Architectural Goals and Constraints](#), to see our performance goals.

Quality

Our main quality goal is to be fun and easy to use, but here are more concrete goals:

Scalability: We don't expect to have many users, but if it ends up being too much for our resources, we'll probably just prevent new users from signing up, and only users with accounts can use the app.

Availability: The app can be downloaded from GitHub. It won't be on the Android or Mac app stores. It will be available for both Android and iOS phones.

Portability: It can be used on Android and iOS phones.

Security: Accounts with usernames and passwords stored on our database.

Risks and Mitigation Plans

Hardware Failure: If our computers break when we're developing, it's the computer's owner's responsibility to fix it.

Software Failure: If any of our software doesn't work, we'll just do our best to fix it or find an alternative program.

3 Design Description Organization

Introduction

The Design Description Organization section goes over the details in the structure and functions of the Oxward Fictionary app and its software. It does this through four different descriptions:

1. Decomposition Description: A description of the business functions. This is a map describing what Oxword Fictionary is and what it does. It's useful for a general understanding of the app and the project.

2. Dependency Description: A description of the overall structure of the software specifically. It contains a map of the different screens and what the user will be able to do on each screen.

3. Interface Description: A description of the UI of the app. It only contains a set of goals we have for readability, fun, and intuitiveness.

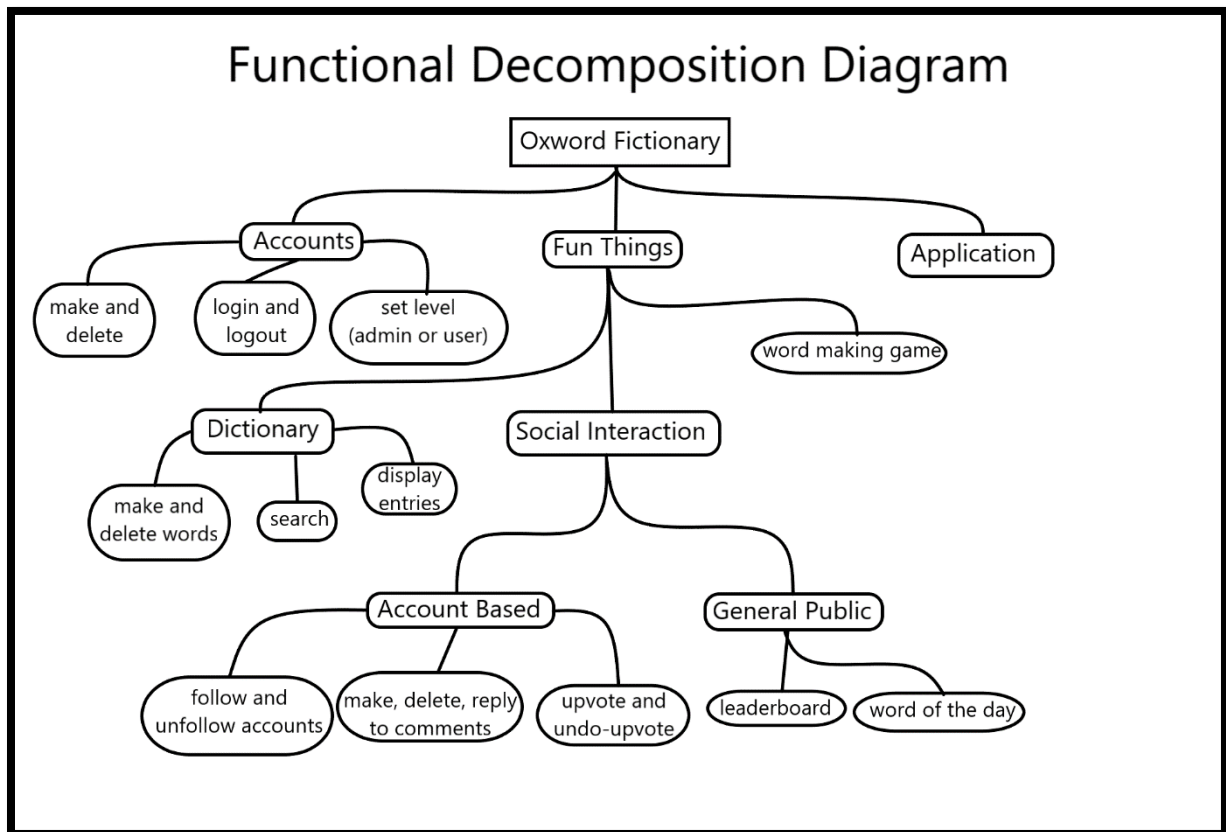
4. Detailed Design Description: A detailed description of how the software will work.

Decomposition Description

Use: The Decomposition description of the project is a view of the business functions of the Oxward Fictionary. It answers the questions "What is Oxword Fictionary?" and "What kinds of things does it do?"

Representation

Here's a diagram to answer that:



The main purpose is to just be fun, and it'll do this first and foremost through the fake dictionary, but it will also have social interaction and word making games to make it more interesting. Besides this, there will also be an important feature of having accounts to allow people to interact with each other, as well as some basic app features such as changing the theme.

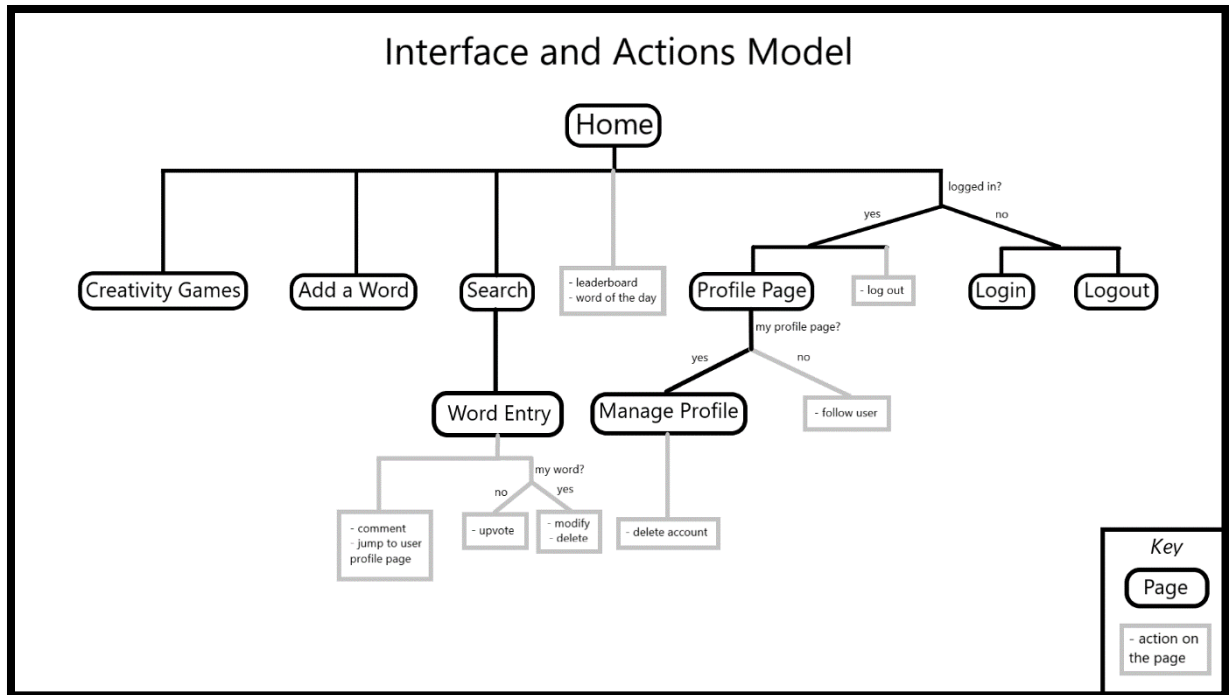
Dependency Description

Use: Dependency Description describes the overall structure of the program.

Representation

This section contains a map of the different pages of the app, and the features and functionality that will be included in each page.

The black circles are different pages, while the grey boxes are the functions that you can do on each page.



Interface Description

Use: Interface Description contains the goals we have for our interface.

Representation

UI Goals:

- **Fun and enjoyable**
 - o People smile and sometimes laugh from using the app/ui
 - o Surprises?
 - o Fun fonts/colors
 - o Don't force it or overdo it
- **Intuitive**
 - o You find things where you expect to find them
 - o Access important/common things with few button presses
- **Readable**
 - o Uniform: everything looks like it goes together
 - o Fonts/colors
 - Big enough to read, and contrasts enough with background to see
 - Fun, but also nice to look at so it doesn't get annoying after awhile
 - o Simple:
 - Elements not cramped

- No unnecessary clutter
- Good transition times (if possible)
 - Instantaneous time: 0.1 seconds
 - Basic action flow: 1 second or less
 - Time for user to lose attention: over 10 seconds
- **Dynamic**
 - Elements show/hide depending on the user and their permissions

Detailed Design Description

Use: The Detailed Design Description contains the logical flow and lifecycle of the program. It's for the programmers, so they have a better idea on how to structure the code.

Representation

Since we are building Oxword Fictionary with Xamarin, the actual program flow and lifecycle is already designed, we just need to implement our specific pages. Here is a basic explanation of how Xamarin apps work:

1. The application starts.
2. The main page is loaded.
 - a. UI is rendered from View files (.XAML).
 - b. Logic is handled with ViewModels file (written in C#).
3. The main page loads other pages, which have their own Views and ViewModel files, which can load other pages, or terminate, etc.
4. Sometimes the application is put to sleep when the user focuses on another app, or the phone is put to sleep.
5. When the user opens the app when it is asleep, it resumes.

We don't need to implement any of this except the View files and ViewModels because the hard workers at Microsoft already did most of the backbone work for us when they made Xamarin.

Here is a diagram of the things we will implement:

