{EPITECH INNOVATIVE PROJECTS}

La Vie est un Jeu !

Technical Documentation

CONFIDENTIAL summary of the technical components of the project for developers

CONFIDENTIAL

This document is CONFIDENTIAL and internal for the project « Life Is A Game ».

Without a special agreement, the only persons authorized to read this document are the official members of the current project and the team of professors of the EIP Laboratory (Epitech Innovative Projects) at Epitech.

The only persons authorized to give exceptional permission or share this file are project managers of the project « Life Is A Game »

Team

Lepage Barbara lepage.barbara@gmail.com
Caradec Guillaume guillaume.caradec@gmail.com
Corsin Simon simoncorsin@gmail.com
Glorieux François fra.glorieux@gmail.com
Klarman Nicolas nickoas@gmail.com
Lassagne David david.lassagne@gmail.com
Louvigny Guillaume guillaume@louvigny.fr
Le-Cor Wilfried wilfried.lecor@gmail.com
Lenormand Frank lenormf@gmail.com

Abstract

"Life Is A Game" offers mobile and tablets applications, and websites to transform your life into a game in which you challenge yourself with goals and reward badges.
The "Specifications" document provides all additional information.

Developers are divided into teams within the group.
We use a number of tools for organizing grouped on a portal, including agendas weekly meetings, repositoryions and Gitub "from" Github to manage tasks.

Development teams correspond to different projects constituting our project :

respecting a Web service API :
This web service is used by mobile clients, tablets, and web External to manipulate the database. It is developed with **Javascript Node.JS** and **MongoDB**x.

Mobile and Tablets :
These applications are available on Android, iOS and Windows Phone through technology **Mono**.

Website :
The website can be used on mobile, tablet or "desktop". It developed with **Ocsigen** in **OCaml**.

Documents :
Many dot our documentation project, such as documentation called "global" (requested by the lab EIP) in **LaTeX** "internal" (corresponding to repositories) in **Markdown** or "code" using the **Doxygen**.

This documentation is not sufficient to itself alone, developers should take the time to read all the documentation related.

Document Information

Document Type	Technical Documentation
Full title Document	Technical Documentation « Life Is A Game » for new developers
Keywords	« Technical », « Developer », « API » « lavieestunjeu »
Number of pages	20
Manager	Group Leader : Barbara Lepage
Group Members	See cover page
Authors	Group members, see cover Page
Contact	lavieestunjeu@googlegroups.com
Current revision	1.0
Vitrine website	http://eip.epitech.eu/2014/lavieestunjeu/
Official Website	Not available

Revisions table

Version	Auteur	Commentaire	Date
0.1	Barbara Lepage	Layout	15/03/2013
0.2	Barbara Lepage	Introduction	3/15/2013
0.3	Barbara Lepage	Tools and Requirements	15/03/2013
0.4	Barbara Lepage	API	15/03/2013
0.5	Barbara Lepage	Mobile and tablet	20/03/2013
0.6	Barbara Lepage	Website	20/03/2013
0.7	Barbara Lepage	Documentation	20/03/2013
0.8	Barbara Lepage	Conclusion	3/20/2013
0.9	Barbara Lepage	Summary document	3/20/2013
1.0	Barbara Lepage	First version	3/20/2013

Table des matières

Chapitre I

Introduction

This document is intended for new developers joining the team development of the project « Life Is A Game ».
We assume that readers of this document have already read The document "Specifications" and programming bases.

Our application is divided into several components :

- A public developer API

- Mobile applications and tablets

- A website

It is not necessary for you to read the entire document if you do not participate as a component, but we recommend to be in line with the rest of the team and have a global vision the project.

Chapitre II

Tools and prerequisites

II.1   Portal

As we are many developers, we require an organization copy to complete our project. To do this, we use portal which includes all of our tools.

The portal is available at the following URL : http://life.paysdu42.fr/
It is necessary for all developers on the team to get to all days of work on the project.

The sources of the portal is available at the following URL : https://github.com/LaVieEstUnJeu/Internal-Tools
More information about repositories in the "Sources" projects.

II.1.1   Organization

The tools we use are all on the portal, but go far in their use allows more efficiency. We set an overview of current tasks in creating new or expanding in need deadlines.

Calendar

Our agenda is hosted by **Google Calendar**. It is strongly recommended to add in your own calendar. UNIT ICAL that Google offers is compatible with most managers calendar market.

We also use a global agenda which corresponds to our GANTT.

Regular Meetings

We organize regular meetings, including a technical meeting a week, a meeting brainstorm inside the application and per week meetings monthly or bi-monthly lab organized by the EIP (Communication, monitoring, technical defenses, defense communication).

During the **weekly technical meeting**, we discuss our advancements in the respective project and then we split by teams for greater efficiency.

Communication

Our **mailing list** lavieestunjeu@googlegroups.com is our main communication tool (in French), but Regarding the technical side of our project MUST be on From GitHub (see below) and written in English.

We also communicate regularly GTalk, IRC, phone and Google Hangout.

Server Development

As some technologies that we use, particularly Ocsigen do are not portable and are very difficult to install, we use common development server on which are installed software and libraries necessary for developers.

II.2 Sources of the projects

II.2.1 Git

All our projects are versioned using the release manager **Git**.

Many references exist on the Internet for these notions, including Wikipedia page version management and official website Git.

II.2.2 GitHub

The host of our Git repositories called **GitHub**. This service offers free public repositories and private repositories paying as well as many other features.

The main features that we use are :

Organizations :
The "organisms" GitHub repositories are used to group on the same page and manage teams.
Our "organization" : https://github.com/LaVieEstUnJeu.

Issues :
The "issues" can be compared to "tickets". They allow to identify and discuss the features under development, bugs, ask questions, suggest new features or manage "pull-request" (addition of external requests changes to the main repository).

II.2.3 Access to repositories

The majority of our repositories is private. To access them, you must be a member team development project.

If you are not yet a member teams allowing you to have access repositories, apply as soon as possible.
Meanwhile, in order to allow new developers who have not yet access to our repositories (and

lab EIP) to browse the contents of our repositories, we create a user account **lavieestunjeu-watcher**. The password for this account is "2h&1H52Z".

The current teams are :

Watchers :
People outside the project with special permission for **see** some repositories (eg in the context of jury EIP official advisers, lab EIP).

Web Service API :
Developers of Web Service API compliance.

Mobile :
Developers of mobile applications and tablets.

Website :
Developer Web Client.

II.2.4   Repositories Terms of Use

General Rules

- The contents of private repositories is confidential and must not not be released.

- The content of public repositories are protected by their licenses respective open-source, directly present in the repositories.

- To facilitate the integration of new members of nationalities different, we use the **English** on these repositories.
  No word French or another language is not tolerated, either the sources in the "issues" or "commit messages."

- Each repository and each sub-project in the repositories should contain a **README**. See section "Documentation" for more information.

"Commit"

- commits responding to an "issue" after the name with a "#"

- commit messages should always be explicit

- It is recommended to segment the commit messages with a summary of less than 50 letters followed by two newlines and a complete description (usually a list)

- A commit must contain one and only one feature

- If the repositories contain several projects it is imperative to put a "tag" at the beginning of the commit message
  Example : "[Portal] Add labels and sub-labels in the menu"

- "diff" commit should never contain lines off-topic.

"Issues"

- There must always be an "open" outcome for each task in

- "The issues contain tags" :

  - task

  - bug

  - bonus

  - issue

- We do not use issues in public repositories

Chapitre III

Developer API Public

III.1   API

The interraction between the data and we handle our services is strictly defined by an API.

III.1.1   Specifications

REST Architecture

This API uses the standardized architectural style **REST** ("Representational State Transfer").

Many references exist on the Internet for these notions, including Wikipedia page.

Formatting JSON

Data published by the API are formatted using the format standardized data **JSON** (JavaScript Object Notation).

Many references exist on the Internet for these notions, including official website.

III.1.2   API Documentation

Our API is constantly evolving, unlike this documentation is why it is not directly transferred to it.

It is available on Google Docs : `http://goo.gl/uxMoJ`.
When our service is published, the API will be available on our website Web

III.2   The Web Service

We offer our users a web service API respecting their to manipulate data services. Our customers and web mobile are also based on Web Service for handling data.

For more information on what a Web service, many references exist on the Internet for these notions, including Wikipedia page.

III.2.1   Sources

Programming Languages

Our Web Service is implemented in **Javascript** the framework **Node.js**.

The choice of this technology is detailed in the specifications.

Many references exist on the Internet for these notions, including Javascript Wikipedia page and official website Node.js.

Repositories

The GitHub repository of Web Service https://github.com/LaVieEstUnJeu/API
It is important to take familiarized the information provided in the "README".
More information about repositories in the "Sources" projects.

III.2.2   Database

We use a database called "no" relational called MongoDB.

Many references exist on the Internet for these notions, including Wikipedia page and NOSQL official website MongoDB.

The database evolves over the progress of the project and we API Modified regularly throughout the development phase.

Conceptual design of the database is available in the document architecture project : https://github.com/LaVieEstUnJeu/Doc/tree/master/doc/AA

The current version - but not final - the database is MongoDB available on the repository : https://github.com/LaVieEstUnJeu/API/blob/master/public/models.js.

III.3   Using the Web Service

Our Web Service is used by our customers and web and mobile potentially by other developers in the future. For this reason, we decided to publish the source code using the API then subsequently propose other examples so that users be guided at best.

The repository **public using the API** : https://github.com/LaVieEstUnJeu/Public-API/
It is important to take familiarized the information provided in the "README" and "all other README" available examples for each entry provided.
More information about repositories in the "Sources" projects.

The source code for these examples is available ONLY on the repository to not to dupplication. For this reason, using the repositories code should be calling upon installation.

Chapitre IV

Tablets and Mobile Applications

IV.1   Sockets and technology

Our apps are available on mobiles and tablets, several Operating Systems

To minimize the amount of code generated and thus limit the amount of potential bugs, we use technology **Mono** C#.
In particular, we use :

- Native code C# to **Windows Phone 7**

- Mono.android for **Android**

- MonoTouch for **iOs**

The choice of this technology is detailed in the specifications.

Many references exist on the Internet for these notions, including documentation Official C# and official website Mono.

IV.2   Source

IV.2.1   Repositories

The GitHub repository applications : https://github.com/LaVieEstUnJeu/Applications
More information about repositories in the "Sources" projects.

IV.2.2   Unit Tests API

Applications filing also proposes to implement unit tests client-side API.

IV.3   Design

The design of the views and the description of the features thereof defined in this document are CONFIDENTIAL : http://goo.gl/oY9se

Chapitre V

Website

V.1   Technology

The website is developed with **Ocsigen**, server and web framework in **OCaml**.

The choice of this technology is detailed in the specifications.

Many references exist on the Internet for these notions, including official website OCaml and official website Ocsigen.

Client code and server code are developed with the same technology. The views are responsives, ie the display is automatically adapted to display on said "desktop environment", "mobile" or "tablets".

V.2   Repositories

The GitHub repository website : https://github.com/LaVieEstUnJeu/Website
It is important to take familiarized the information provided in the "README".
More information about repositories in the "Sources" projects.

V.3   URLs

URLs are defined on the "README" : https://github.com/LaVieEstUnJeu/Website

V.4   Views Design

The design of the views defined in this album CONFIDENTIAL : album.

V.5   Features

The description of the features on the views defined in the "specifications".

Chapitre VI

Documentation

This chapter is intended for technical writers project.
In the section "code" is open to all developers.

VI.1   Overall project documentation

VI.1.1   Sources

The GitHub repository overall project documentation : https://github.com/LaVieEstUnJeu/Applications
It is important to take familiarized the information provided in the "README".
More information about repositories in the "Sources" projects.

VI.1.2   Steps and technologies

We document our projects according to the wishes of the lab and add EIP also other documentation we deem useful.

Writing

If necessary, we write the content of our documents using a collaborative tools (Google Docs / Drive) allowing us to work and more distance.

If only one person working on the document, it is not necessary to use this tool and we directly use the layout tools.

Layout

We set our page documents using **LaTeX** a markup language.

Many references exist on the Internet for these notions, including the Wikipedia page and official website LaTeX.

VI.2   Documentation directly related to repositories

Each repository and each sub-project in the repositories should contain documentation in README "file".

VI.2.1   Content

This file should contain :

- content, the purpose of repository / project

- The project dependencies

- A simple and quick to install, compile, run, test

- The tree and the location of the main

- An explanation for people who join the project

- "coding-style" (or coding convention) Project

- A FAQ

- The copyright (for repositories open-source)

VI.2.2   Sources

These documents must be written using the language markup **Markdown** (recommended) or another language supported by GitHub.

The list of languages supported by GitHub and links to their documentation are available on GitHub documentation.

VI.3   In the code

It is important to comment your code and documenting best. To do this, keep in mind that other people will need ironing behind to read, understand and modify, and they must do so without difficulty and without the help of developers Original

We generally recommend using the syntax **Doxygen** to automatically generate documentation, but each project has its own coding conventions defined in the "README".

Many references exist on the Internet for these notions, including Wikipedia page coding conventions and official website Doxygen.

Chapitre VII

Go further

This document is a simple guide for new developers join the project without difficulty, but it is not him only complete documentation.

It is therefore essential that as developers, you take the time to read about the projects you are working on using the links in this document and doing your own research.

It is also important for the cohesion of the group to communicate within teams. For this, participation in meetings is mandatory and can ask questions or answer.

Finally, it is important that you take the time, as did and make the team, write documentation for your projects respective and holding them up to date.