

Software requirement specification (SRS) document template

Project name: Weather Cat for University Student

Date: 11/06/2022

Version: v.0 (Origin)

By: Wildcats (CS Group 6)

Revision history

Version	Author	Verson description	Date completed
v.0 (Origin)	Saplyn	Filled the blank	11/06/2022

Review history

Reviewer	Version reviewed	Signature	Date

Approval history

Approver	Version approved	Signature	Date

Table of contents

1

Introduction

- 1.1 Product scope
- 1.2 Product value
- 1.3 Intended audience
- 1.4 Intended use
- 1.5 General description

2

Functional requirements

3

External interface requirements

- 3.1 User interface requirements
- 3.2 Hardware interface requirements
- 3.3 Software interface requirements
- 3.4 Communication interface requirements

4

Non-functional requirements

- 4.1 Security
- 4.2 Capacity
- 4.3 Compatibility
- 4.4 Reliability
- 4.5 Scalability
- 4.6 Maintainability
- 4.7 Usability
- 4.8 Other non-functional requirements

5

Definitions and acronyms

1 Introduction

Describe the purpose of the document.

1.1 Product scope

List the benefits, objectives, and goals of the product.

Helping students to make better use of weather information.

1.2 Product value

Describe how the audience will find value in the product.

Help students to organize their lives better and avoid trouble by giving them weather information and suitable advice.

1.3 Intended audience

Write who the product is intended to serve.

University Students. It includes...

- Sports guys
- Girls who like to make up
- Geography majored students

1.4 Intended use

Describe how will the intended audience use this product.

- Offer weather Info & weather forecast
- Offer activity advices

1.5 General description

Give a summary of the functions the software would perform and the features to be included.

- Weather Info
- Weather Forecast
- Activity Advisor
- Academic Assist (Premium Feature)

2 Functional requirements

List the design requirements, graphics requirements, operating system requirements, and constraints of the product.

Design Requirements

- No specified requirements

Graphics Requirements

- Integrated GPU should be fine
- OpenGL 3.0 or higher

Operating System Requirements

- Android 4.0 and above
- HarmonyOS 2.0.0 and above

Other Constraints

- RAM: 4.0 GB or more

3 External interface requirements

3.1 User interface requirements

Describe the logic behind the interactions between the users and the software (screen layouts, style guides, etc).

Effortless. There should be no impossible-to-press buttons, and the layout should be user-friendly.

Accessible. Everyone should be able to access the app, including the disables (Colorblind).

Personal. The user can customize the app's look to meet their appetite.

3.2 Hardware interface requirements

List the supported devices the software is intended to run on, the network requirements, and the communication protocols to be used.

Supported Devices: Huawei p30 pro, Xiaomi Redmi Note 7, Google Pixel 5...

Network Requirements: IPv4 & IPv6

Communication Protocols: TCP/IP & HTTP/HTTPS

3.3 Software interface requirements

Include the connections between your product and other software components, including frontend/backend framework, libraries, etc.

Framework: Electron.js

Libraries: JavaScript Standard Libraries, CSS Standard Libraries, ElectronJS Libraries

3.4 Communication interface requirements

List any requirements for the communication programs your product will use, like emails or embedded forms.

Email address (for dev contact)

Embedded Form (for bug report & user feedback)

4 Non-functional requirements

4.1 Security	Include any privacy and data protection regulations that should be adhered to.
User Location. It should be protected and deleted after used.	
4.2 Capacity	Describe the current and future storage needs of your software.
Current Storage: 400MB (for app essentials & weather history) Future Storage: less than 800MB (for premium user account info & other features)	
4.3 Compatibility	List the minimum hardware requirements for your software.
RAM: 4.0 GB or more GPU: Integrated should be fine	
4.4 Reliability	Calculate what the critical failure time of your product would be under normal usage.
It should be robust. Normally there should not be any crashes.	
4.5 Scalability	Calculate the highest workloads under which your software will still perform as expected.
It should be robust. Normally there should not be any crashes. And Normally there is no high workloads either.	
4.6 Maintainability	Describe how continuous integration should be used to deploy features and bug fixes quickly.
ElectronJS Buildin updater. Bug fixes count as updates.	
4.7 Usability	Describe how easy it should be for end-users to use your software.
It should be very easy. Nearly no instruction is needed.	
4.8 Other	List any additional non-functional requirements.
None.	

