Grab&Run;

Software Requirements Specification

By:
Marwa Hany
Menna Allah Medhat
Shereen Mohammed Zeinah
Toka Mohamed Naguib

Table of Contents

Preface

Document purpose

The purpose of this document is to give a detailed description of the requirements for the “xx” software. It will explain the purpose and features of the system, the interfaces of the system, functional and non-functional requirements , the constraints under which it must operate. It also serves as the input for design and development.

Document Conventions

- Font type: Calibri.
- Bold words with large font size more than 14 is used for the headlines of the chapters and sections or to concentrate on an important word.
- Words with all characters are Capital refers to an abbreviation.
- Website: italic underlined text with blue color refers to website address or email address.
- sentences between to parentheses ( ) refers to definition or give more information about the previous words. .

Intended Audience and Reading Suggestions

- Developers who can review project’s capabilities and more easily understand where their efforts should be targeted to improve or add more features to it.
- Project testers can use this document as a base for their testing strategy as some bugs are easier to find using a requirements document.
- End users of this application who wish to read about

Revision History

Version	Change reference
0.1	Adding Preface
0.2	Adding Introduction and Glossary
0.3	Adding System Users
0.4	Adding User requirements
0.5	Adding System Architecture and Non functional requirements
0.6	Adding System interfaces and System modeling
0.7	Adding Work break down structure and appendices and reference
0.8	First release
0.9	Modify user interfaces.

1. Introduction

1.1 Survey

There are a lot of educational websites for kids in different subjects as

Nowadays you can find many websites that supports learning of coding with a huge variance. But let's talk a close look on their specs and what they offer, you can easily find websites that supports coding through courses as code avengers, edx, coursera and this is great but unfortunately many of them needs fees and also you will notice that many of them will be hard on kids and students who are not majored in coding.

You can also see many websites that offers the idea of drag and drop of some blocks that symbolizes blocks of codes, this is a great start for beginners in coding as it eliminates complexity of coding syntax ,so the user to this kind of software just drag and drop some predefined blocks to build up his program, as we talk you can notice some disadvantages in this way, first he can`t customize his code, he just use some predefined blocks, second problem he will never get to know real coding or get into the real world of coders he will be always stuck in dragging and dropping, this kind of websites can be found in scratch as user of this website will learn the algorithm of coding without knowing code syntax itself as we mentioned .

We can also take a look on the budget for these websites we will find many of them are offered for fees or trials only so it targets small segment as code avengers not only the budget but also all these software are offered through web applications and this might be a problem to many of us in Egypt and offline software are much preferable.

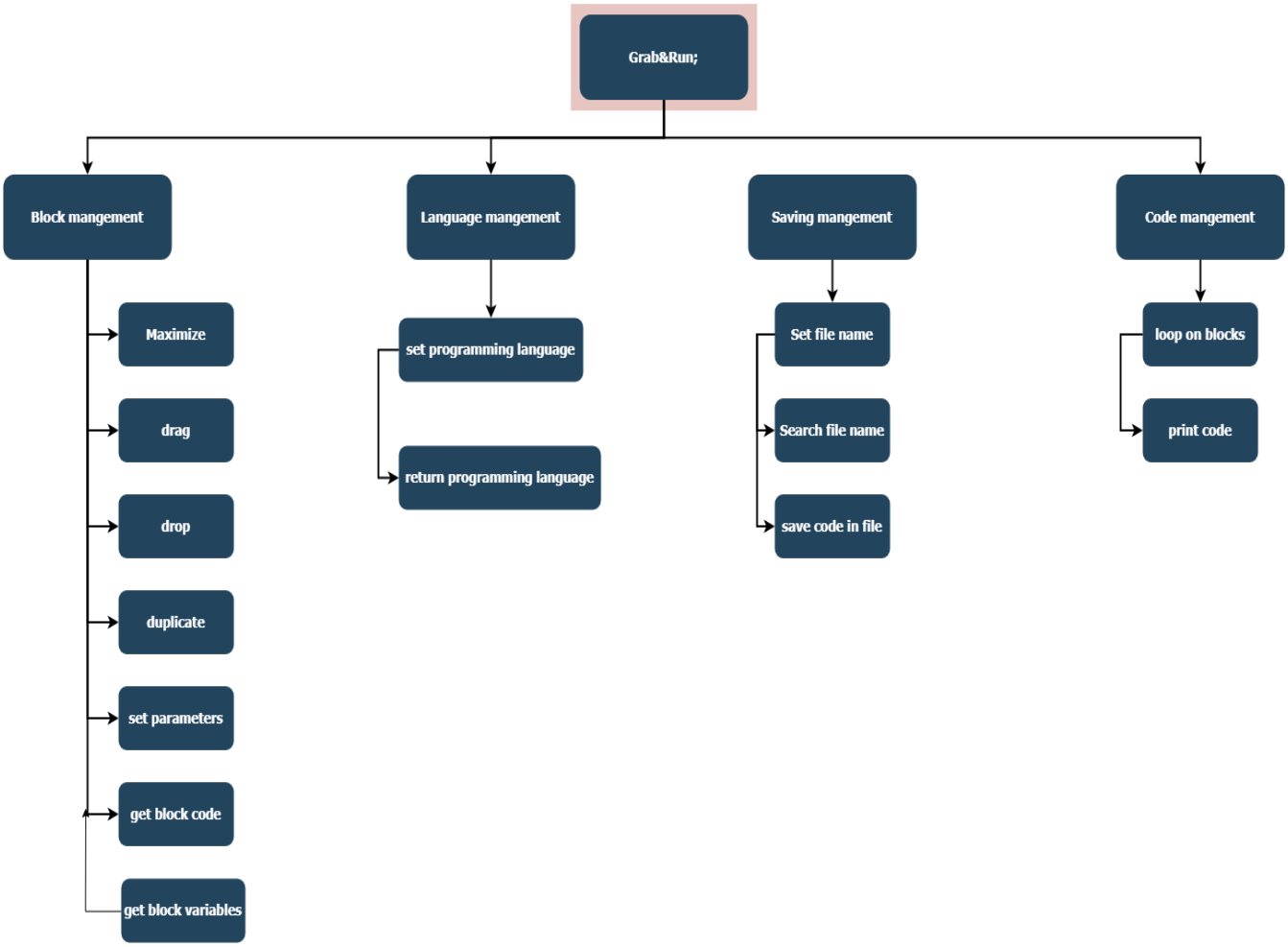
1.2 Purpose

This software is intended to kids and non-majored students to help them to learn programming, algorithm and how to think in a sequential way through a simple easily used software. So to be precise it will enable user to create programming projects through:

- Drag a block the symbolizes some block of code.
- Drop it to their project to create a meaningful algorithm.
- Check this sequence of blocks in a programming language to get introduced to the syntax.
- Enable users to save their projects to use them later.

1.3 Scope:

Facilitation of the process of education through an offline software loaded on our PC and smart phones without the need to have an internet connection. Users will easily download the desktop application and right away use it without the need to any license. It will have a very simple user interface to enable users to use it without the need to any guide. It focuses on logical thinking and programming through, also users don`t necessary have a background of programming and algorithms.



1.4 Similar System

- Code.org - **https://code.org**
- Scratch - https://scratch.mit.edu/
- Code academy - **https://www.codecademy.com**
- Tynker - https://www.tynker.com
- Code avengers - https://www.codeavengers.com/

2.Glossary

2.1 Acronyms, definitions and abbreviations

- Users: kid, Person who are not majored in coding.
- SDD: Software design document.
- SRS: Software requirements document.
- WBS: Work break down structure.

3. System Users

3.1.System stakeholders

System Analyst

- Responsible for requirements gathering.
- Responsible for deployment and support.
- Create SRS.

System designer

- Receive SRS.
- Create SDD.

System testers

- Test the functionalities of the code.

System developers

- Implement and maintain Software based on SDD.

System Users

Learner

- Create a project.
- Explore each block in the displayed blocks.
- Drag blocks into its destined place.
- Select the desired language to be able understanding more than one language.
- See the generated code and try to understand It.
- Different blocks could be dragged.
- Can link between blocks by adding blocks inside some other blocks.
- Can remove a certain block form the drop space.

3.2 User objectives

Learner

- Learn from home or form any desired place**.**
- Be interested in coding.
- Learn how to code in an easy, effective way.
- Know about different functionalities in programming should be available in different blocks.
- Internet connection is not a must to learn something like coding.

- Self-Learning.

4. User requirements definitions

4.1 System functions

Select block

- User has the ability to select any block from blocks placed in the selection zone .

Drag block

- User can drag the block from the selection zone.

Drop block

- User can drop the selective block in the droppable zone.

Rearrange blocks

- User can easily rearrange blocks after he put them in droppable zone to change the sequence of his program.

Set parameters

- User can interact with a block by setting its parameter, to make the code more specific and meaningful.

Link blocks

- User can link between blocks to make meaningful sequence of these blocks.

Reuse blocks

- User can instantiate the same block if he need to use it in another statement of code.

Choose programming language

- User will have the option to choose which programming language he wishes his code appear with as python , c++, java .

Save code

- User can save his code. so he can return to it any times he wants.

Generate code

- System can translate the sequence of blocks into a specific code with the chosen programming language.

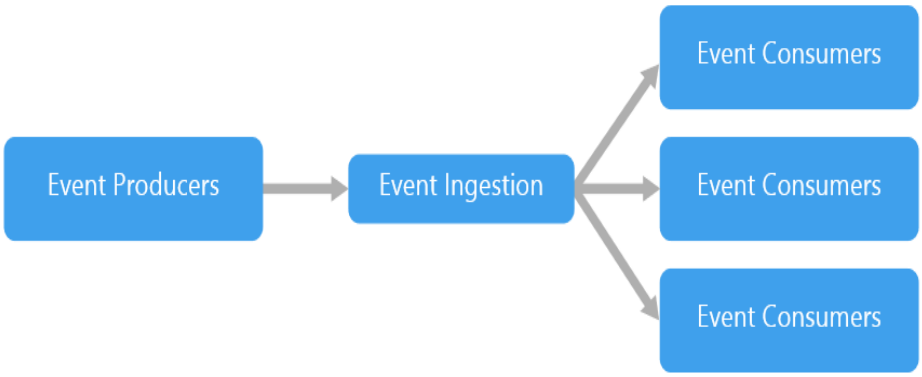
4.2 Constraints

- All the components should be draggable and droppable.
- components should be instantiated after each drag , and they should be scalable.
- after adding the component into the scene it should translated into code in simple way and in correct syntax .
- dragged blocks should not be dropped in any panel except droppable panel.
- Certain blocks has its type of inputs .
- Remove the code generated after removing its corresponding block.

5. System architecture

Event-driven architecture

An event-driven architecture consists of event producers that generate a stream of events, and event consumers that listen for the events.



Events are delivered in near real time, so consumers can respond immediately to events as they occur. Producers are decoupled from consumers — a producer doesn't know which consumers are listening. Consumers are also decoupled from each other, and every consumer sees all of the events.

6. Non functional requirements

6.1 Scalability

- the system consists of components which is loosely coupled and tightly cohesive so adding new features will be possible and uncomplicated task.

6.2 Installability

- the system can be installed, uninstalled, or reinstalled easily into a target environment.

6.3 portability

- An end-user is use this system on any OS either it is Windows or Linux.

6.4 availability

- system is available 24 hours in day and 365 day in year as system is a desktop application only need computer device to operate on it.

6.5 usability

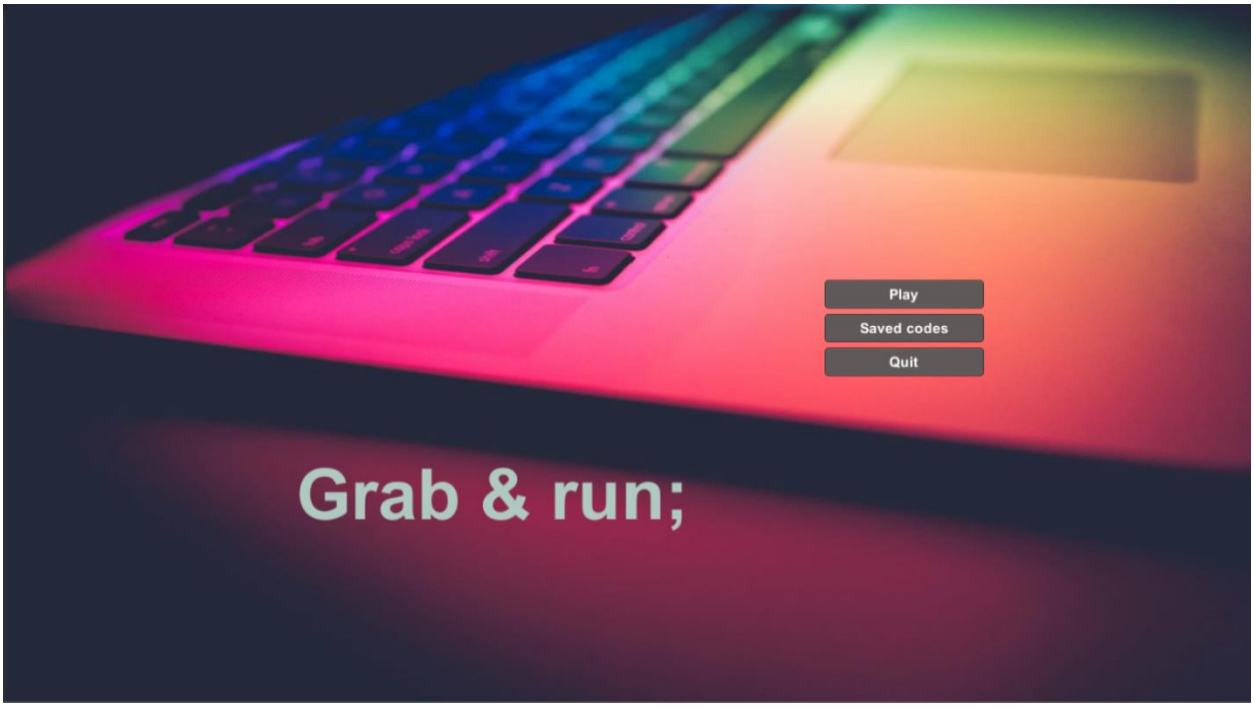
- using a meaningful graphics that will help users to understand and interacting with system easily

6.6 synchronization

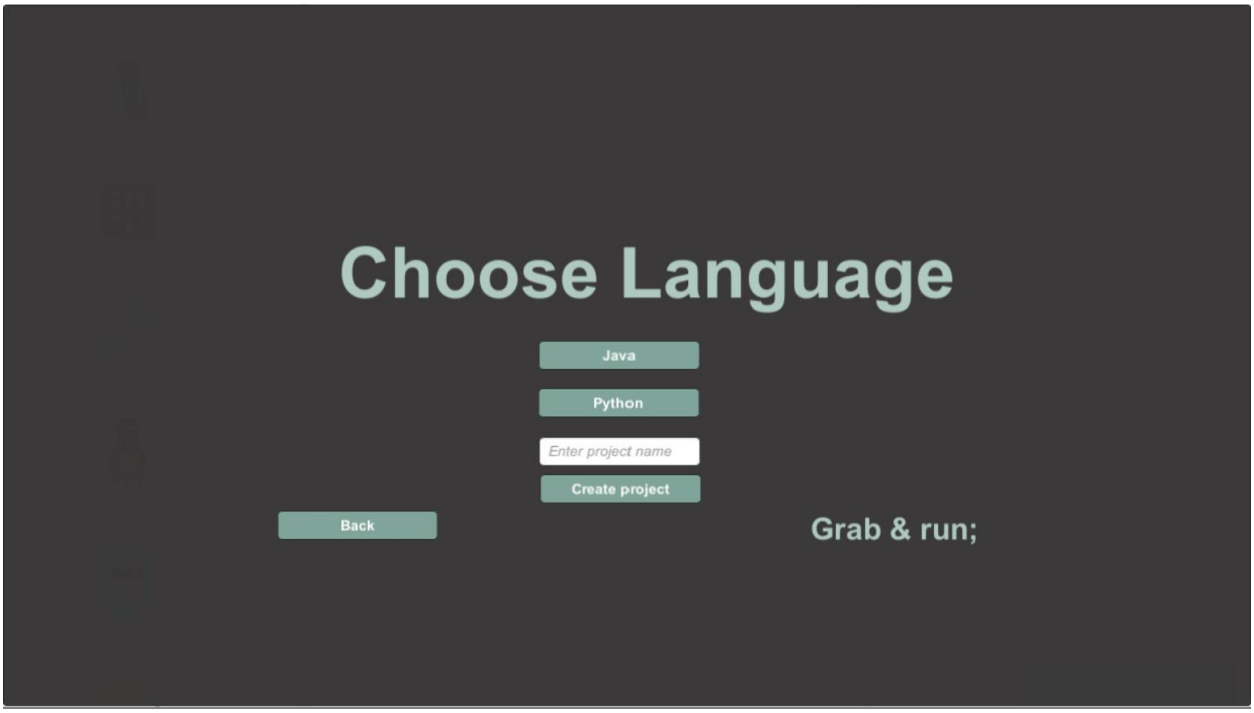
- u code is appear at the same time block dropped in droppable panel.

7. System interfaces

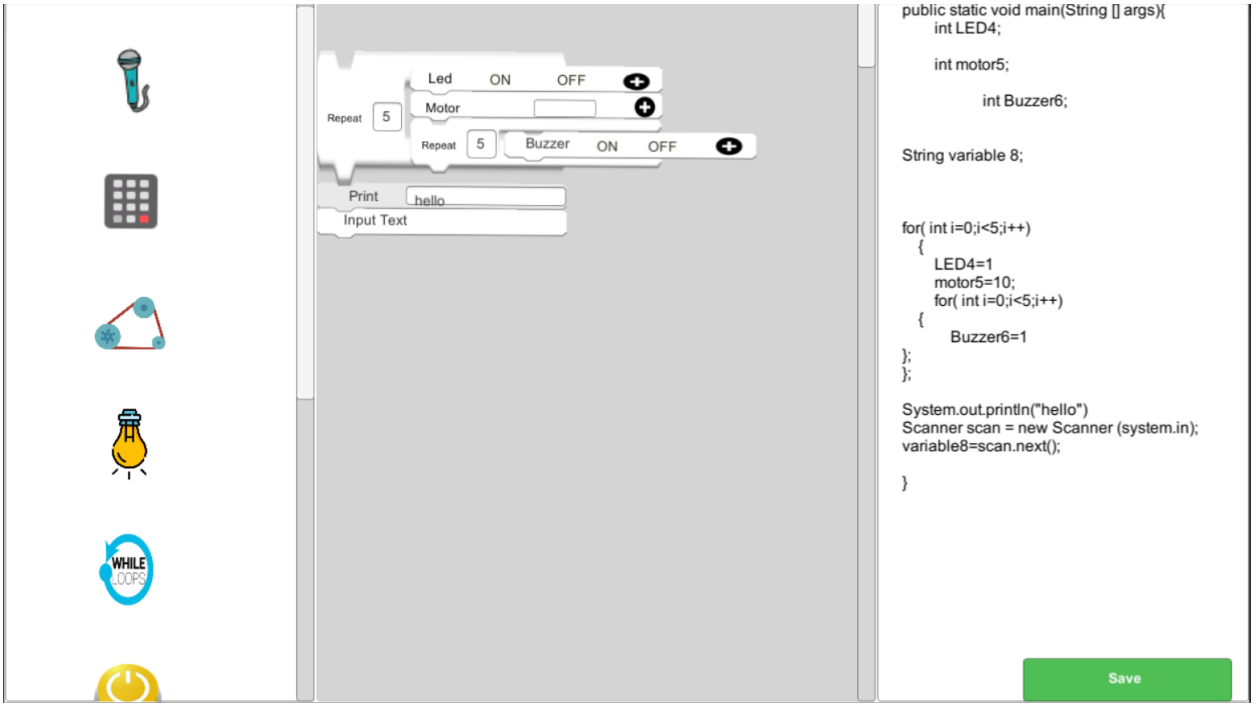
7.1 User Interfaces



play scene



choose language scene



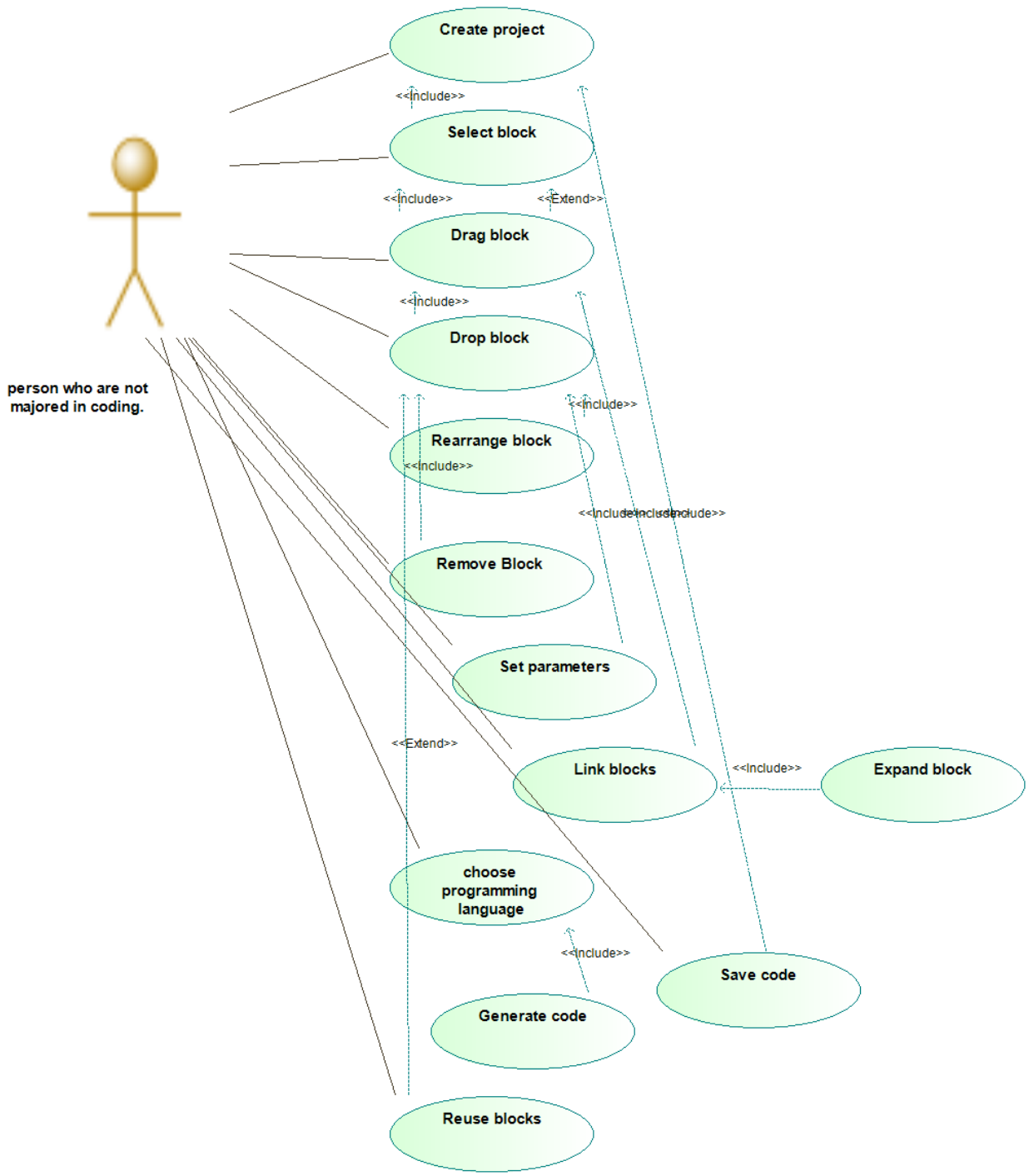
Main scene

7.2 Software interfaces

- Operating System : Unix, Linux, Mac, Windows etc..
- Development tool : Unity , C sharp

8. System Modeling

8.1 Use Case Diagram

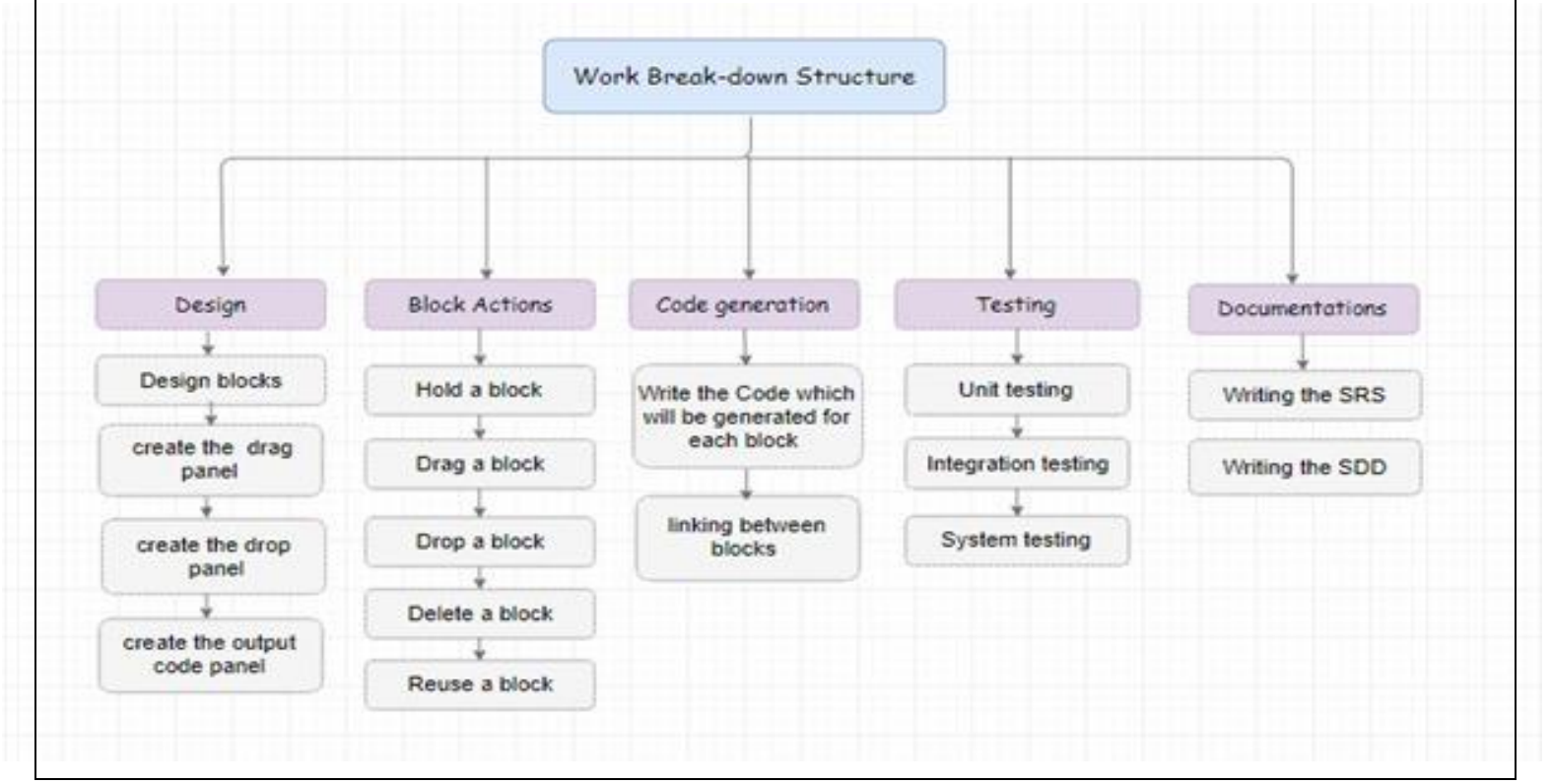


9. System Evolution

Version number	Features added to the system
1.1	Maintain the faults of the previous version of system according to the rating forms of users reviews.
1.2	Adding more blocks to cover more of the problem solving ideas and programming language syntax.
1.3	Adding a stimulation feature so user can test his code after finishing it.

10.Work plan

10.1 Work break down structure (WBS)



10.2 Gantt Chart

11. Appendices

1- High quality images for diagrams & charts. Folder Name: “Diagrams ”.

12. References

Similar Systems:

Code.org - **https://code.org**
Scratch - https://scratch.mit.edu/
Code academy - **https://www.codecademy.com**
Tynker - https://www.tynker.com
Code avengers - https://www.codeavengers.com/