

Requirements and Analysis Document for Group 04

Version: 1.2.0

Date: 18/5-17

Author: Hanna Carlsson, Tobias Lindgren, Sara Kitzing, Madeleine Lexén

This version overrides all previous versions.

1 Introduction

This application's purpose is to get more young people interested in programming and learning how to code. To make the whole process less daunting, and to give an easy start to the journey. The issue at hand is the high threshold to begin coding as a complete beginner. It can be hard to get started and there is a growing demand for programmers. The point is to develop younger people's interest in programming, and give them a way to get started.

The beneficiary of the application will be youth who might be interested in learning how to code. In the long term it will also benefit companies and educational institutions within computer science. The application will primarily be used as an educational tool during leisure time, but it does not hinder it from being used in school as well.

The application will be similar to a game, as in that it will have several levels in a hierarchy where the user will have to complete a series of steps to reach the next level. These steps will be in form of questions or code assignments.

1.2 Definitions, acronyms and abbreviations

Key: The correct answer

Level: A subcategory within a main category. Contains information about the specific topic and a question on the topic.

Map: The start page, a map over the different main categories

Read more: A page which shows information about basic programming in text form that the user will learn throughout the different worlds. It can be accessed from the menu in all views.

Boss: The last level in a main category which summarizes what the previous levels has taught the user.

Fill in the blanks: A question where the user will get a number of sentences with blanks that they will have to fill in themselves.

Write code: A question where the user will get a task where they need to write code

Toast: A rectangle with text that appears at the bottom of the page

2 Requirements

2.1 User interface

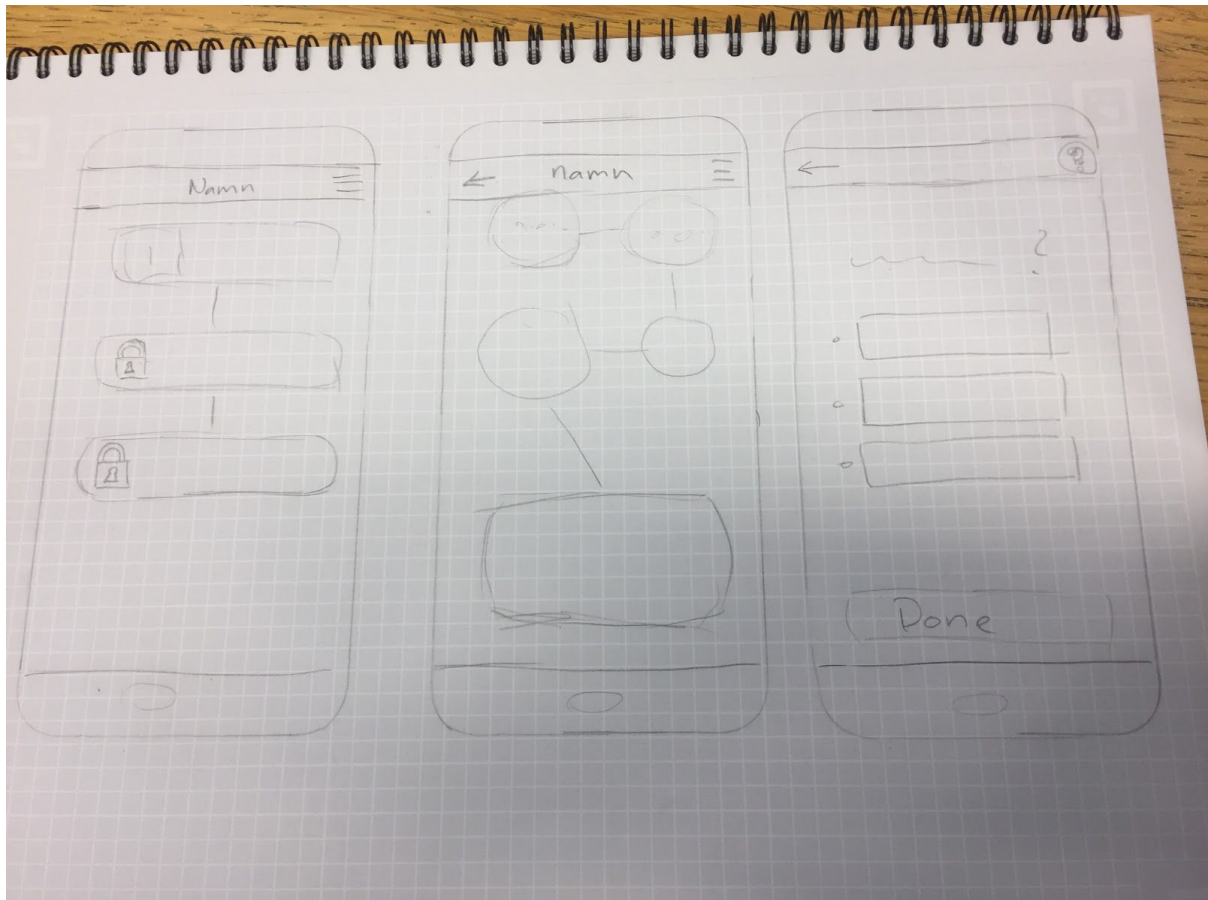


Figure 1. The left view is the start page of the app. Shows the main map of the different worlds. The middle view is the levels view, shown when clicked on a world. This shows the different steps you need to complete to get to the next world. The right view is shown when clicked on a level. This displays a question you need to answer correctly in order to pass the level.

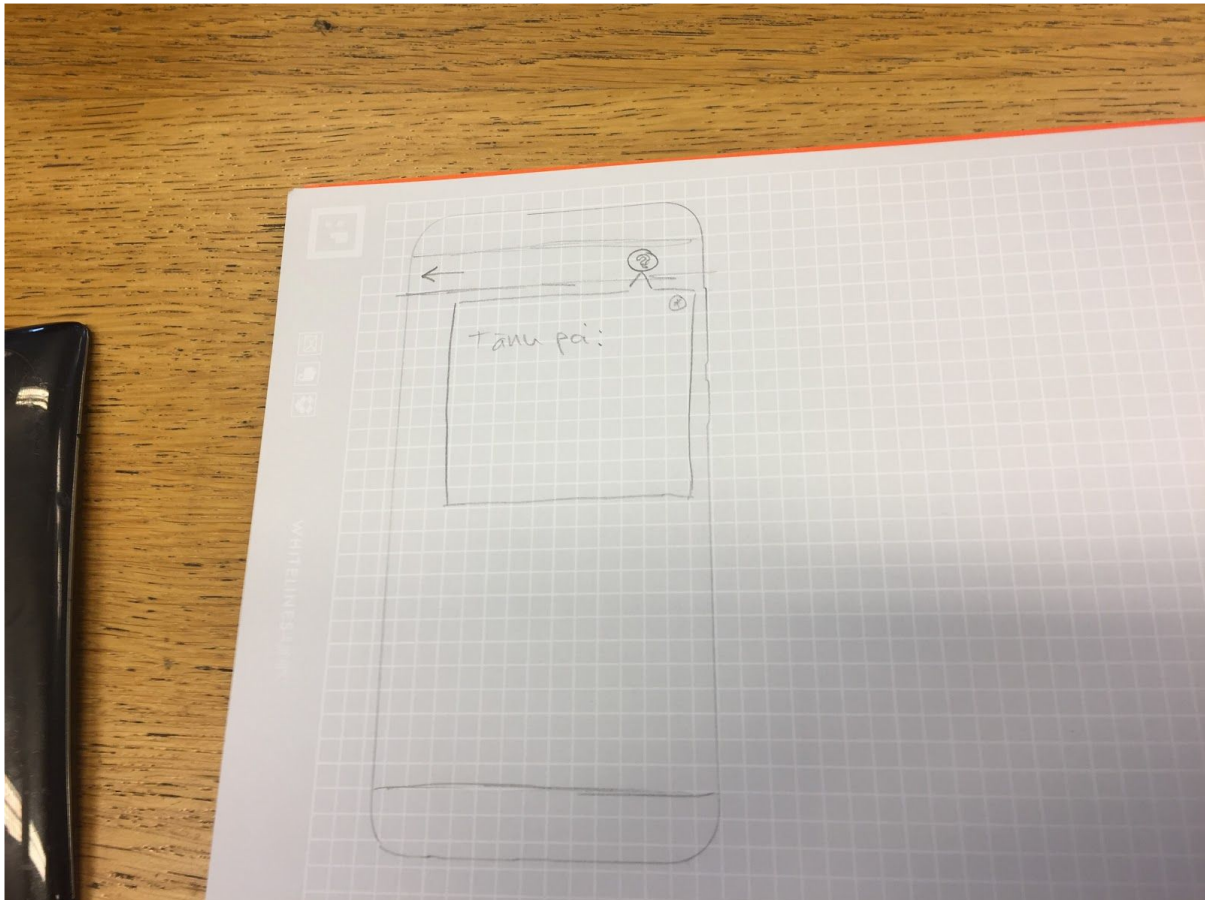


Figure 2: The hint view. Shown when you click on the question mark in the right corner.

2.2 Functional requirements

The user starts out with only the first category unlocked, the user can then advance through the main categories by unlocking them. This is done by completing every level in previous main categories, which unlocks the next main category. The levels themselves consist of four assignments and at the end of the level there is a boss that the user has to pass. The test is another question but this time you have to write actual java code.

If the user struggles with the answer to an assignment there are two hint, that guides the user to the answer. If the user still isn't able to find the right answer, there will be an option to show the key to the assignment.

The assignments are constructed accordingly; first there is an explanation of the issue at hand, then follows a question testing if the user understood the information. These test can take on one of three forms, which are fill-in-the-blanks, multiple choice and write code.

There will also be a section where the user can read more about what it is learning about, with a vocabulary and texts explaining the subjects further.

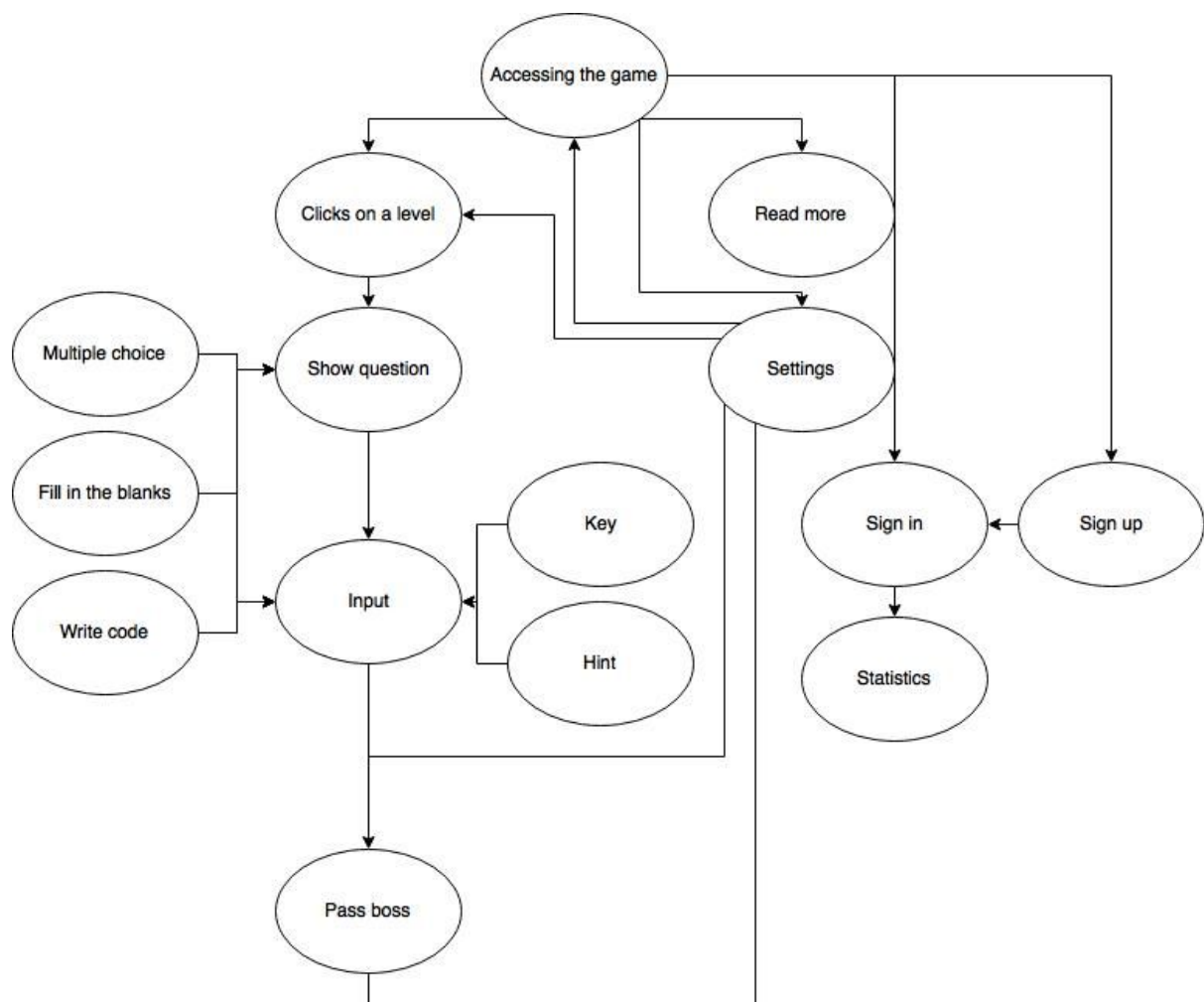
1. Accessing the game
2. Input
3. Hint
4. Show the correct answer

5. Passes the boss
6. Read more

2.3 Non-functional requirements

The application should be able to run on all android smartphone devices which run on an OS Lollipop 5.0 or higher.

3 Use cases



3.1 Use case listing

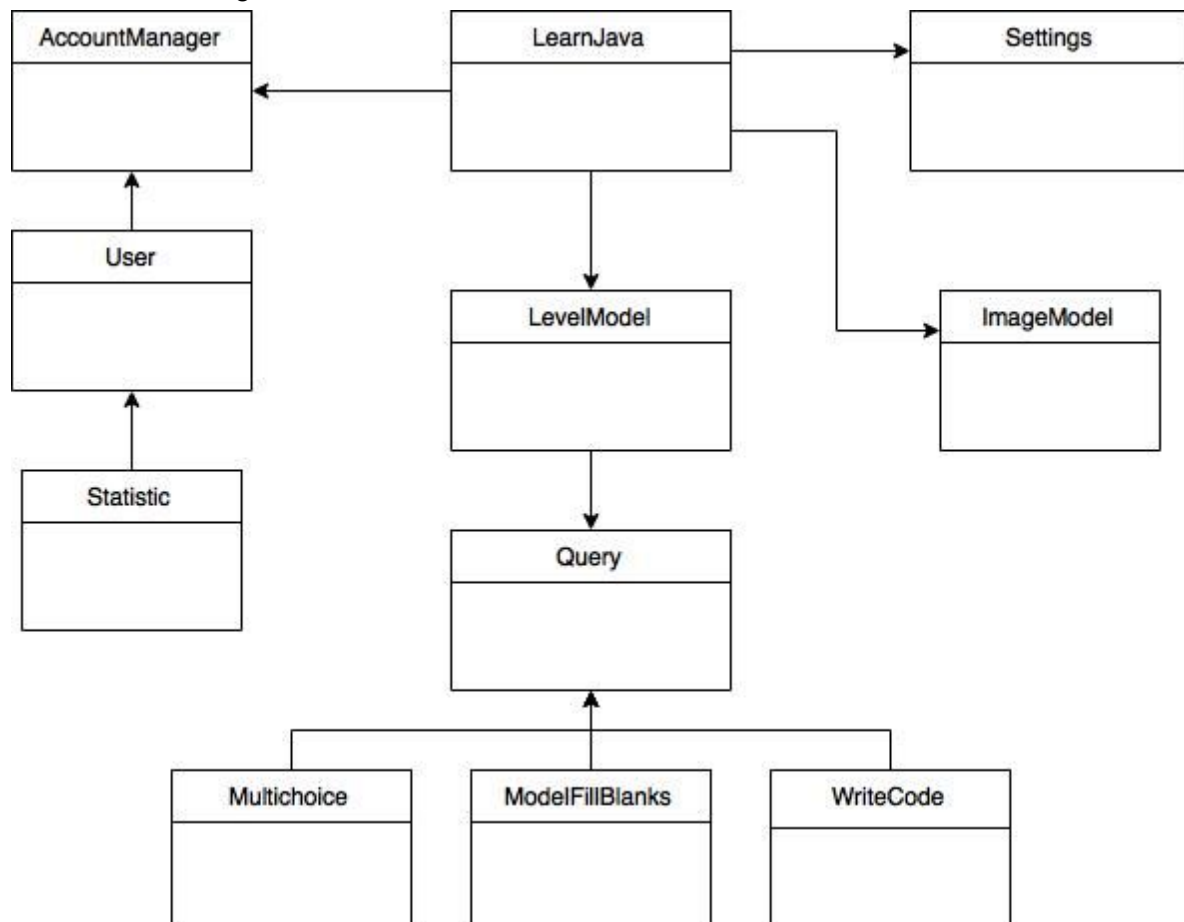
- Accessing the game
 - Sign in
 - Sign up
 - Clicks settings
 - Clicks about appen

- Click Read more
- Click statistics
- Play game
 - Check answer
 - Fill in the blanks
 - Multiple question
 - Write code
- Check answer
 - Passes the boss
 - Write code check
 - Show hint and key

The full user cases can be found in the appendix

4 Domain model

An UML class diagram .



4.1 Class responsibilities

LearnJava: This class is responsible for maintaining and storing the hashmap with the level objects.

LevelModel: This class is responsible for keeping track of the questions, and maintaining them. It also creates the

different query objects, each level model object has a query object as an instance variable.

Query: This class is an abstract superclass to the question classes, and gives them a common interface.

Multichoice: This class handles the logic for the multichoice question, and corrects the user's answer.

ModelFillBlanks: Handles the query where the user gets to fill in the words that are missing in a text.

WriteCode: Handles the query where the user gets to write its own code.

AccountManager: Handles the log in and add user functions along with loading and storing the user objects.

User: Handles the updating of the user and holds a statistics object which calls the save statistics methods.

Statistics: Stores the time it takes to finish the question, how many hints needed and if you peeked at the key.

ImageModel: Saves and loads the profile pictures of the users.

Settings: Handles the different settings available

5 References

6 Appendix

6.1

Use case: Accessing the game

Summary: The user opens the app and logs in

Priority: high

Extends:

Includes: Sign in, Sign up, clicks settings, clicks statistics, clicks about appen, clicks read more

Participants: The user, the system

Normal flow of events

The player opens the app and signs in/signs up.

	Actor	System
1	Opens the app	
2		Shows the login page
3.1.1	Write username and password	
3.1.2		(Password is correct) Logs in and shows the start page, the main categories
3.1.3		(Password is incorrect) Toast shows the password or username is incorrect
3.2.1	Clicks on sign up	
3.2.2		Shows the sign up page
3.2.3	Writes username, password and chooses a profile picture	
3.2.4		(Username is ok) Logs in and shows the start page, the main categories
3.2.5		(Username if already taken) Toast shows the username is already taken
4		Shows main category view

Alternate flow

Flow 2 (Check settings)

	Actor	System
--	-------	--------

1	Clicks on settings	
2		Shows the settings page
3.1.1	Drags volume slider	
3.1.2		Volume changes
3.2.1	Clicks on sounds off button	
3.2.2		Sound stops

Alternate flow

Flow 3 (Clicks read more)

	Actor	System
1	Klicks "Read more"-button	
2		Shows read more page

Alternate flow

Flow 4 (Clicks statistics)

	Actor	System
1	Clicks "Statistics"-button	
2.1		(If logged in) Shows users statistics
2.2		(If not logged in) Toast showing message that you can't access statistics when not logged in

Alternate flow

Flow 5 (Clicks about appen)

	Actor	System
1	Clicks "About	

	appen"-button	
2		Shows information about the app

6.2

Use case: Play game

Summary: The user plays the game

Priority: high

Extends: Accesing the game

Includes: Check answer, Fill in the blanks, Multiple choice, Write code

Participants: The user, the system

Normal flow of events

The user clicks on an unlocked level

	Actor	System
1	Clicks on category button	
2.1		(If unlocked) Shows level view
2.2		(If locked) Nothing happens
3	Clicks on level button	
4.1		(If unlocked) Shows level information
4.2		(If locked) Nothing happens
5	Clicks on continue button	

6		Shows question, Alternate flow 2, Alternate flow 3, Alternate flow 4
7	Clicks on submit button	
8		User case check answer

Alternate flow

Flow 2 (Fill in the blanks)

User is on a fill in the blanks question

	Actor	System
1		Shows question with numbers as blanks
2	Fills in the answer connected to the number	

Flow 3 (Multiple choice)

User is on a multiple choice question

	Actor	System
1		Shows relevant question with multiple choices
2	Selects a radiobutton	

Flow 3 (Write code)

User is on a write code question

	Actor	System
1		Shows assignment
2	Answer the question by writing code	

6.3

Use case: Check answer

Summary: The user enters the an answer and the app acts upon it.

Priority: high

Extends: Play game

Includes: Passes the boss, write code check, show hint and key

Participants: The user, the system, the server

Normal flow of events

The player enters the right answer.

	Actor	System
1		(If no input) Toast shows no input
2.1		(If write code question) Alternate flow 2
2.2		Checks if answer is correct
3.1		(If correct) Shows passed level view
3.1.1	Clicks on next level button	
3.1.1.1		(If write code question) See alternate flow 3
3.1.1.2		See user case play game (4.1)

3.1.2	Clicks on back button	
3.1.3		Shows level map
3.2		(If incorrect) Shows failed level view
3.2.1	Clicks on try again button	
3.2.1.1		(If correct) See user case check answer (2.2)
3.2.1.2		(If incorrect) See alternate flow 4

Alternate flow

Flow 2 (Write code check)

	Actor	System
1		Starts server, sends users code
2		Compiles the code and sends back
3		Checks if answer is correct

Flow 3 (Passes the boss)

	Actor	System
1.1		(If not in category 4) Shows main category

		view, new category is open
1.2		(If in category 4) Shows main category view, toast shows congratulation message
2		See user case play game

Flow 4 (Show hint and key)

The user is unsure of how to solve an objective

	Actor	System
1		Show question view with “?”-button
2	Clicks on hint button	
3.1		(If user is wrong once) Show hint
3.2		(If user is wrong more than once) Show hint and key
5	Clicks on ok	
6		See user case play game (6) but with “?”-button as well