



# **Requirements and Analysis Document for Qwalk**

*Version 2.0*

**2017-05-28**

Chalmers Tekniska Högskola

Elina Olsson, Jonathan Kraft, Pia Westerbergh, Kevin Solovjov

*This version overrides all previous versions.*



<b>1 Introduction</b>	<b>3</b>
1.1 Purpose of application	3
1.2 General characteristics of application	3
1.3 Definitions, acronyms and abbreviations	4
<b>2 Requirements</b>	<b>4</b>
2.1 User Interface	4
2.2 Functional Requirements	7
2.3 Non-Functional Requirements	8
<b>3 Use Cases</b>	<b>9</b>
3.1 Use case listing	9
3.1.1 Use Case 1: Sign in	10
3.1.2 Use Case 2: Choose action	12
3.1.3 Use Case 3: Play a qwalk	14
3.1.4 Use Case 4: Show Map	16
3.1.5 Use Case 5: Update Map	18
3.1.6 Use Case 6: Change game settings	20
3.1.7 Use Case 7: Answer Question	21
3.1.8 Use Case 8: Show Results	23
3.1.9 Use Case 9: Create/edit a qwalk	24
3.1.10 Use Case 10: Add option question	25
3.1.11 Use Case 11: Add tiebreaker Question	26
3.1.12 Use Case 12: Set question location	27
3.1.13 Use Case 13 : Add friend	28
<b>4 Domain Model</b>	<b>29</b>
4.1 Class Responsibilities	29



# 1 Introduction

## 1.1 Purpose of application

The project aims to create a location based game as an android application where the user can create and play premade qwalks (the official name for quiz walk in this project) by using the GPS function on a mobile device. The idea of the game originates from the concept of a classic quiz walk (*tipsrunda* in Swedish), but in a digital interpretation. The users will be able to play alone or against a computer opponent.

## 1.2 General characteristics of application

The application will run on android mobile devices in portrait mode, using a graphical user interface.

The quiz walks takes place on a digital map generated by Google Maps, where every question has a geographically bound position. The application is GPS-based which lets the player navigate on the digital map by walking around in reality. The player has to be within a predetermined area around the question to be able to answer it.

The user can choose to play different challenges, such as competing against a computer opponent, hiding questions or showing them all at the same time.

It is also possible to add other users as friends in order to gain access to their quizzes. In order to support this functionality, the application will also need to communicate with a database on a server. This database also stores account information, and if a user doesn't have an account they will not be able to create their own, or get access to friends' qwalks . However, any user can play a premade qwalk, and they have the possibility of creating an account directly in the application.



## 1.3 Definitions, acronyms and abbreviations

**Qwalk** - is the name of the application. The word is a composition of the words *quiz* and *walk*.

**qwalk** - is the same word as the application Qwalk, but starts with a lowercase q. The word refers to an actual quiz walk and not the application.

**Monkey** - is a computer opponent in the shape of a monkey. The player can choose to challenge the monkey on three different levels.

**Arrow** - an arrow that points in the direction of the next question.

**Map** - the actual world map and also the game board.

## 2 Requirements

### 2.1 User Interface

The graphical user interface in the application is based on the sketches below.

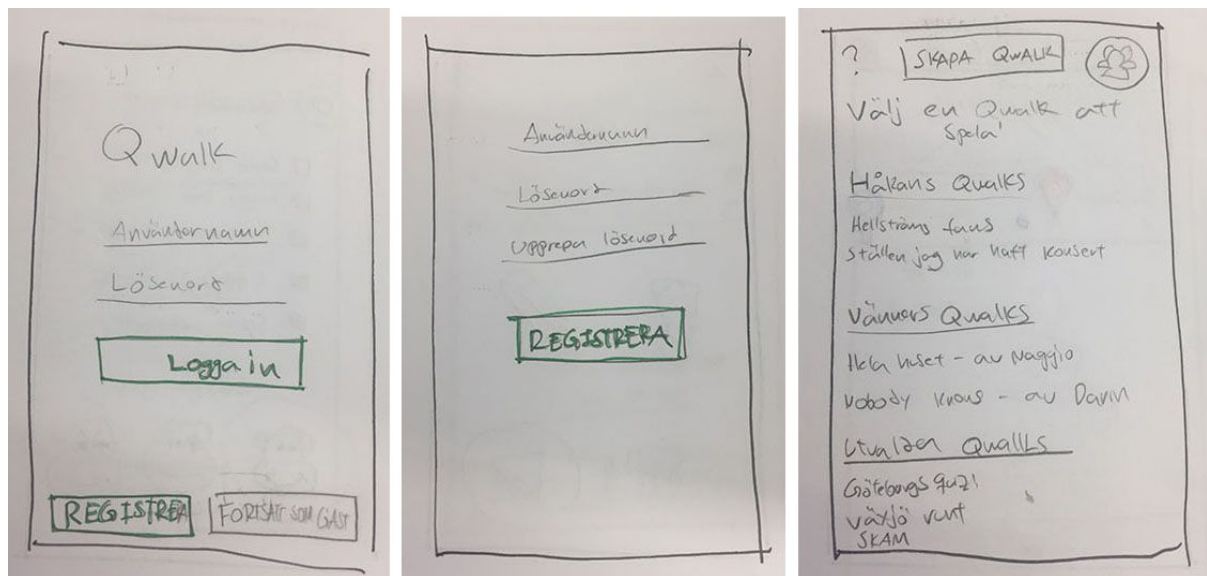


Fig. 1 - The first menus for signing in, register and overview of available qwalks.

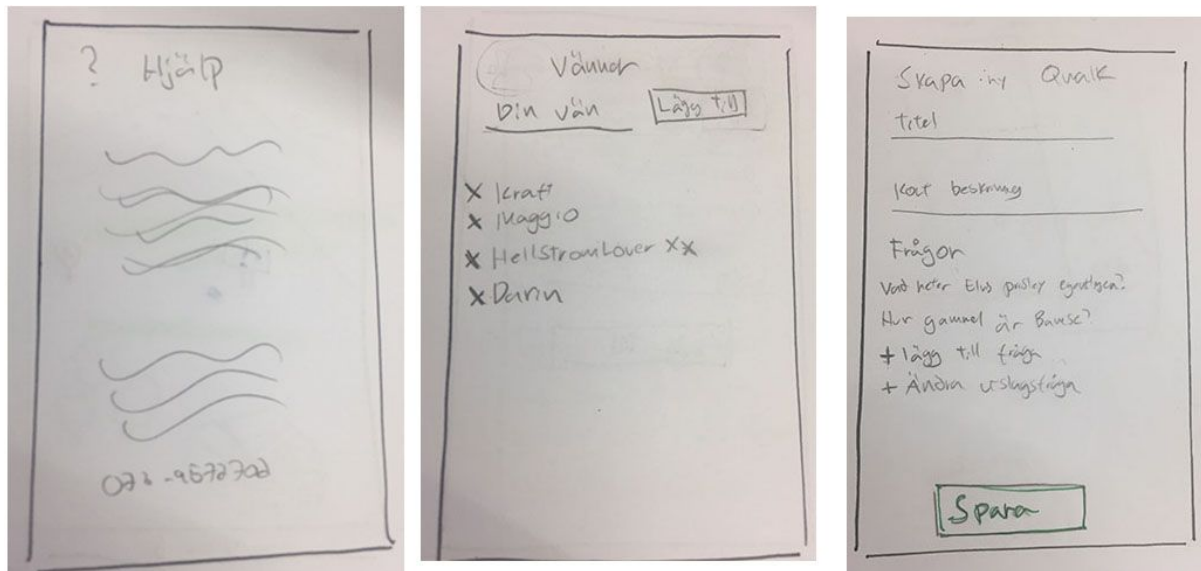


Fig. 2 - Views for help, add friends and create new qwalk.

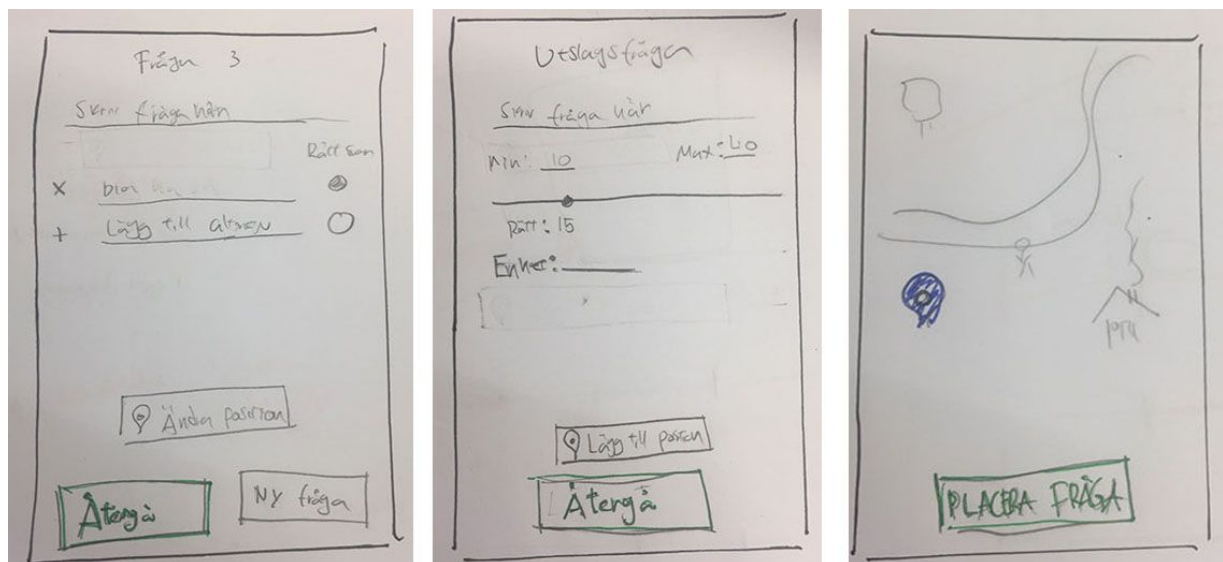


Fig. 3 - Views for creating and placing option questions and tiebreaker

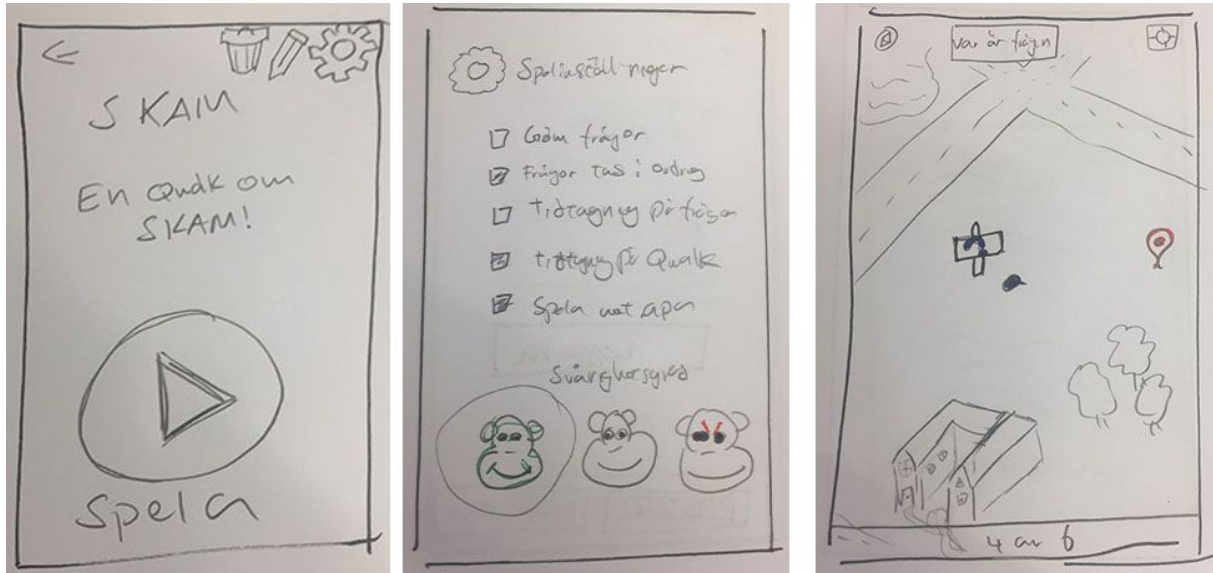


Fig. 4 - Detailed view for playing a qwalk, game settings view and the view for finding a question on the map.



Fig. 5 - The first two pictures show the map view and answer view when challenging a monkey. The third view is for answering a tiebreaker.



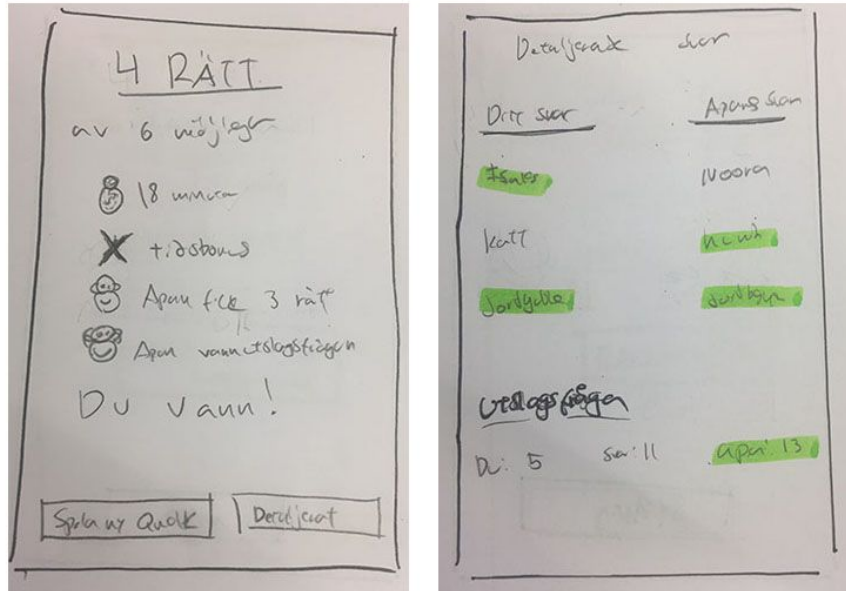


Fig. 6 - Views for presenting the result of a finished qwalk.

## 2.2 Functional Requirements

The player should be able to:

1. Play a qwalk
  - a. Play with different settings
    - i. Hide all questions
      1. Only see the questions if they are in range
    - ii. Answer the questions in order
    - iii. Set a game time limit
    - iv. Set a question time limit
    - v. Challenge a monkey on three different difficulties
      1. See the monkey moving on the map
      2. See the monkey's answers and result
  - b. See current position
  - c. See questions markers on a map
  - d. See the direction to the closest question
  - e. See if a question is in range
  - f. Answer a question
  - g. See how many questions the player has yet to answer
  - h. See results after a finished qwalk
2. Create or edit a qwalk
  - a. Add option questions
    - i. Add position
  - b. Add tiebreaker
    - i. Add position



3. Register an account and sign in
4. Add other users as friends and get access to their qwalks
5. Remove a qwalk

## **2.3 Non-Functional Requirements**

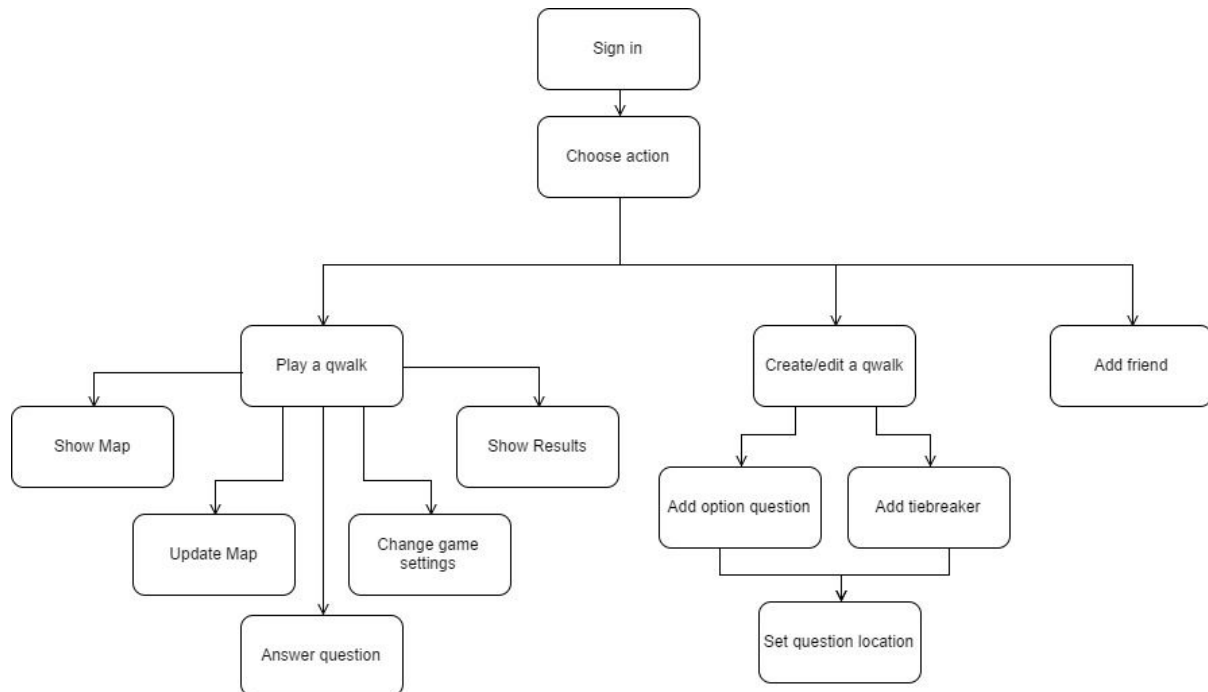
One of the main non-functional requirements is the ability to test the application throughout the development. Other requirements includes a simple design, a responsive feeling and should also considered easy to navigate. The application does not need to have permission to use images and other resources created by others, since the application will only be used privately. Lastly, the software should be somewhat robust with a goal to have no application-terminating errors.





## 3 Use Cases

This is an overview for all use cases.



### 3.1 Use case listing

All use cases for the application Qwalk are listed below.

1. Sign in
2. Choose action
3. Play a qwalk
4. Show Map
5. Update Map
6. Change game settings
7. Answer Question
8. Show Results
9. Create/edit a qwalk
10. Add option question
11. Add tiebreaker
12. Set question location
13. Add friend



### 3.1.1 Use Case 1: Sign in

**Summary:** The user starts the application and signs in to get access to the main functionalities. If the user does not have an account they can choose to register or play as guest.

**Priority:** High

**Extends:** None

**Includes:** UC 2: Choose action

**Participators:** Actual player

#### Normal flow

	User	System
1	Starts the application Qwalk	
2		Shows login view with text fields for entering username, password, and buttons for login, register and guest play.
3	Types in username and password, clicks on login button.	
4		Shows start menu view.

#### Alternative flow 1 - Register

The user registers a new account.

	User	System
1	Starts the application Qwalk.	
2		Shows login view with text fields for entering username, password, and buttons for login, register and guest play.
3	Clicks on register button.	
4		Shows register view with text fields for username, password, confirm password, and a register button.
5	Types in username, password,	



	confirms password and clicks on the register button.	
6		Shows start menu view.

### Alternative flow 2 - Play as Guest

The user does not want to sign in, instead they play in guest mode.

	User	System
1	Starts the application Qwalk.	
2		Shows login view with text fields for entering username, password, and buttons for login, register and guest play.
3	Clicks on play as guest button.	
4		Shows start menu view.



### 3.1.2 Use Case 2: Choose action

**Summary:** The user is on the start menu view and is about to choose where to navigate.

**Priority:** High

**Extends:** UC 1: Sign in

**Includes:** UC 3: Play a qwalk, UC 9: Create/edit a qwalk, UC 13: Add friend

**Participants:** Actual player

#### Normal Flow - Choose to play a qwalk

	User	System
1		Shows start menu view with scrollable lists of the user's qwalks, friends' qwalks and premade qwalks along with other navigation buttons.
2	Clicks on a quiz.	
3		See <i>UC 3: Play a qwalk</i> .

#### Alternative flow 1 - Choose to create a qwalk

Only signed in users can create a qwalk.

	User	System
1		Shows the start menu view with scrollable lists of the user's qwalks, friends qwalks and premade qwalks and a button for creating a new Qwalk.
2	Clicks on the create new Qwalk button	
3		See <i>UC 9: Create/edit a Qwalk</i>

#### Alternative flow 2 - Choose to edit a qwalk

Only signed in users can edit a qwalk.

	User	System
1		Shows start menu view with scrollable lists of the user's qwalks,



		friends' qwalks and premade qwalks along with other navigation buttons.
2	Clicks on a quiz	
3		Shows view for qwalk details of the selected qwalk, with title and description. Also shows buttons for playing qwalk, game settings and if the user owns the qwalk; editing and deleting it.
4	Clicks on icon for editing a qwalk.	
5		See UC 9: Create/Edit Qwalk

### Alternative flow 3 - Choose to add friends and their qwalks

Only signed in users can add friends and their qwalks.

	User	System
1		Shows start menu view with scrollable lists of the user's qwalks, friends' qwalks and premade qwalks along with other navigation buttons.
2	Clicks on the friend button.	
3		See UC: 13 Add friend



### 3.1.3 Use Case 3: Play a qwalk

**Summary:** The player has chosen a qwalk to play and wants to start the game.

**Priority:** High

**Extends:** UC 2: Choose Action

**Includes:** UC 4: Show Map, UC 5: Update Map, UC 6: Change game settings, UC 7: Answer Question, UC 8: Show Results

**Participators:** Actual player

#### Normal Flow

	User	System
1		Shows view for qwalk details of the selected qwalk, with title and description. Also shows buttons for playing qwalk, quiz settings. If the user owns the quiz, there are also buttons for editing and deleting it.
2	Clicks the change settings button.	
3		See <i>UC 6: Change Settings</i> .
4	Clicks the play qwalk button.	
5		See <i>UC 4: Show Map</i>
6		See <i>UC 5: Update Map</i>
7	Walks towards the question marker.	
8		Repeat from <i>step 2</i> until the player is within the range of a question.
9		Changes the question mark (?) on the question marker to an exclamation mark (!).
10	Clicks on the question marker.	
11		See <i>UC 7: Answer Question</i>
12		Repeat from <i>step 1</i> until all questions in the quiz are answered or until the user ends the game.



13		See UC 8: Show Results
----	--	------------------------





### 3.1.4 Use Case 4: Show Map

**Summary:** The system shows the map with the user's current position as well as question markers, buttons and quiz progress.

**Priority:** High

**Extends:** UC 3: Play a qwalk

**Includes:** None

**Participators:** System

#### Normal Flow

	System
1	Shows the map view with question markers for available questions, indicator of the user's position, quiz progress and an direction arrow if the question mark is offscreen.

#### Alternative flow 1 - Hidden Questions Challenge

If Hidden Questions challenge is enabled, do the following.

	System
1	Shows the map view with the indicator of the user's position and quiz progress. A question marker is visible when the user is in range of a question.

#### Alternative flow 2 - Speed Challenge

If Speed Challenge is enabled, do the following.

	System
1	Show map view, pins of all enabled questions, avatar and zoom in on current position and a timer.

#### Alternative flow 3 - Monkey Challenge

If Monkey Challenge is enabled, do the following.

	System
1	Shows the map view with question markers for available questions,



	indicator of the user's position, quiz progress and an direction arrow if the question mark is offscreen and a monkey.
--	--



### 3.1.5 Use Case 5: Update Map

**Summary:** The system updates the map and the location of both the user's position indicator and the monkey, it also shows pins, buttons and timers etc. Combined flows are possible if different settings are selected in the same qwalk.

**Priority:** High

**Extends:** UC 3: Play a qwalk

**Includes:** None

**Participants:** System

#### Normal flow

	System
1	Updates the direction of the arrow which points to the next question, moves the indicator for the player's position to the user's current position and centers the view on the indicator.

#### Alternative flow 1 - Hidden Questions Challenge

If Hidden Questions is enabled, do the following.

	System
1	Moves the user indicator to the player's current position and centers the view on the avatar.

#### Alternative flow 2 - Speed Challenge

If Speed Challenge is enabled, do the following.

	System
1	Updates the direction of the arrow that points to the next question, moves the avatar to the player's current position and centers the view on the avatar. Updates the timer.

#### Alternative flow 3 - Monkey Challenge

If Monkey Challenge is enabled, do the following.

	System
1	Updates the direction of the arrow that points to the next question,



	moves the user position indicator to the player's current position and centers the view on the avatar.
2	Moves the monkey in the direction of the next question until the monkey reaches the last question or until the player has answered all questions.



### 3.1.6 Use Case 6: Change game settings

**Summary:** This is how the user changes the game settings for a qwalk before playing it.

**Priority:** Medium

**Extends:** UC 2: Choose Action

**Includes:** None

**Participants:** Actual player

#### Normal Flow

	User	System
1		Shows the game settings view. There are checkboxes for following settings: <ul style="list-style-type: none"><li>• Hide questions</li><li>• Take questions in order</li><li>• Time limit for qwalk</li><li>• Challenge a monkey</li></ul>
2	Clicks on preferred settings.	
3		Ticks the clicked checkboxes.
4	Clicks the return button.	



### 3.1.7 Use Case 7: Answer Question

**Summary:** The user is within the range of a question and is now about to answer a question.

**Priority:** High

**Extends:** UC 3: Play a qwalk

**Includes:** UC 4: Show map, UC 8: Show results

**Participators:** Actual player

#### Normal Flow - Answer option question

	User	System
1		Shows the view for answering an option question. Shows the question, the answer options and a disabled answer button.
2	Clicks on the chosen answer.	
3		Changes color on the button for the chosen answer and enables the answer button.
4	Clicks on the answer button.	
5		See <i>UC 4: Show map</i> .

#### Alternative flow 1 - Answer tiebreaker

	User	System
1		Shows the view for answering a tiebreaker. Shows the question and a slider for choosing the correct answer.
2	Drags the finger over the slider until the slider displays preferred value.	
3		The value on the slider changes.
4	Clicks on the answer button.	
5		See <i>UC 8: Show results</i> .



### Alternative flow 2 - Time Challenge

If Time Challenge is enabled, do the following.

	User	System
1		Shows the question view, the title of the question, the question, the answer options and a timer in the upper right corner.
2	Clicks on the chosen answer.	
3		Timer stops.
4	Clicks on close button.	

### Alternative flow 3 - Monkey Challenge

If Monkey Challenge is enabled, do the following.

	User	System
1		Shows the view for answering a question. Shows the question, the answer options and a disabled answer button.
2	Clicks on the chosen answer.	
3		Changes color on the button for the chosen answer and enables answer button.
4	Clicks on the answer button.	
5		The button for the monkey's answer gets a stamp with a picture of a monkey and the answer button changes to a close button.
6	Clicks on the close button.	
7		See UC 4: Show map.





### 3.1.8 Use Case 8: Show Results

**Summary:** The user has finished a qwalk and the results are presented.

**Priority:** Low

**Extends:** UC 7: Answer Question

**Includes:** None

**Participants:** Actual player

#### Normal Flow

	User	System
1		Displays the number of correct answers, the total game time, a button for showing a detailed view of the answered questions and a button for playing a new qwalk.
2	Clicks on button for detailed view.	
3		Shows the chosen answers and the correct answers for all questions.

#### Alternative flow 1 - Monkey challenge

Shows the flow if the monkey challenge is activated.

	User	System
1		Displays the number of correct answers for the user and the monkey, the total game time, a button for showing a detailed view of the answered questions and a button for playing a new qwalk.
2	Clicks on button for detailed view.	
3		Shows the user's and the monkey's chosen answers and the correct answers for all questions.



### 3.1.9 Use Case 9: Create/edit a qwalk

**Summary:** Create or edit a qwalk.

**Priority:** High

**Extends:** UC 2: Choose action

**Includes:** UC 10: Add option question, UC 11: Add tiebreaker

**Participants:** Actual Player

	User	System
1		Shows view for creating a qwalk. There are text fields for quiz title, description, a list of created questions and buttons for adding a new question, adding a tiebreaker and for saving.  If editing a qwalk, the fields and lists are already filled in.
2	Fills in title and description.	
3	Clicks on the add option question button, or presses an existing question to edit.	
4		See UC 10: Add option question
5	Clicks on the add tiebreaker question button, or presses an existing question to edit.	
6		See UC 11: Add tiebreaker
7	Clicks on the save button.	



### 3.1.10 Use Case 10: Add option question

**Summary:** This is how the user adds an option question.

**Priority:** High

**Extends:** UC 9: Create/edit a qwalk

**Includes:** UC 12: Set question location

**Participants:** Actual player

#### Normal Flow

	User	System
1		Shows form for creating a new question with a text field for the question and one for an option. There are buttons for adding a position, add a new question and done.
2	Writes a question in the question field.	
3	Writes an answer option in the option field.	
4		Shows a radio button next to the option field and adds a new text field for another option.
5	Repeat from <i>step 2</i> until maximum 4 options are added.	
6	Clicks on add location button.	
7		See <i>UC 12: Set question location</i> .
8	Clicks either return or the new question button.	
		Either return or go back to <i>step 1</i> .



### 3.1.11 Use Case 11: Add tiebreaker Question

**Summary:** This is how the player adds a tiebreaker.

**Priority:** Medium

**Extends:** UC 9: Create/edit a qwalk

**Includes:** UC 12: Set question location

**Participants:** Actual player

#### Normal Flow

	User	System
1		Shows form for creating a tiebreaker with a text field for the question, a slider with a value, text fields for min- and max value, a button for setting location and a done button.
2	Writes the question and min- and max values in the text fields.	
3		Displays the changes in the text fields.
4	Drags the slider to the correct answer.	
5		The slider value changes to chosen answer.
6	Clicks on set location button.	
7		See <i>UC 12: Set question location</i> .
8	Clicks on done button.	



### 3.1.12 Use Case 12: Set question location

**Summary:** The player sets a location for a question on the world map.

**Priority:** High

**Extends:** UC 10: Add option question, UC 11: Add tiebreaker

**Includes:** None

**Participants:** Actual player

#### Normal Flow

	User	System
1		Shows a Map view zoomed in on the user's current position. The user position indicator is in the middle of the screen along with a question marker. There is also a button for placing the question.
2	Clicks on the map or drags the marker to preferred location.	
3	Clicks the place question button.	



### 3.1.13 Use Case 13 : Add friend

**Summary:** The user adds new friends and gets access to their qwalks.

**Priority:** Medium

**Extends:** UC 2: Choose action

**Includes:** None

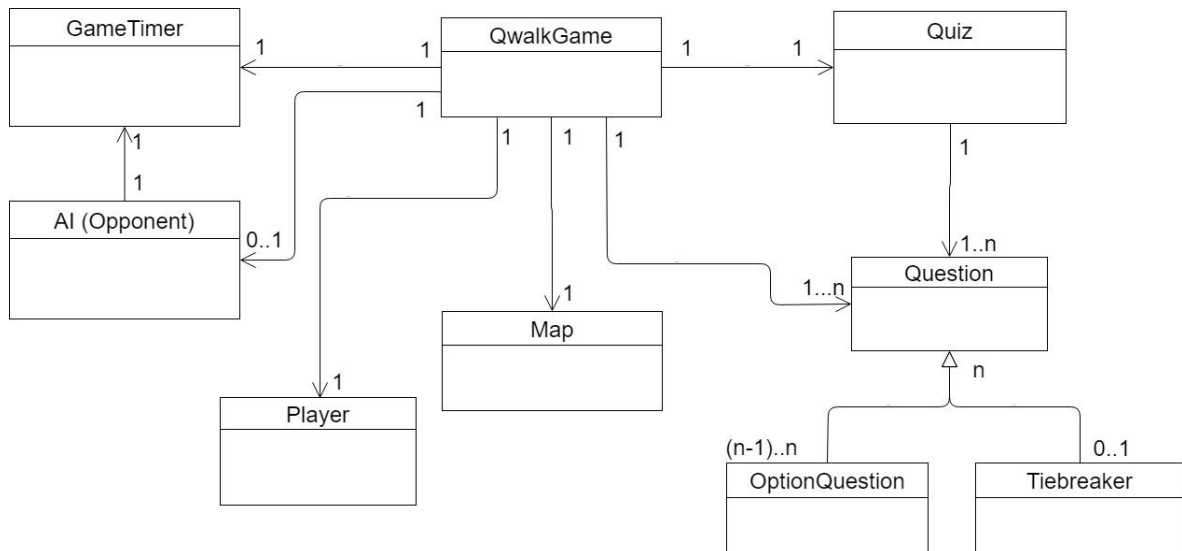
**Participants:** Actual player

#### Normal Flow

	User	System
1		Shows view for finding friends with a list containing the user's friends and a text field and button for adding new friends.
2	Writes the username of the friend the user wants to add.	
3	Clicks on the add friend button.	
4		Adds friend to the friend list and adds all the friend's qwalks in the list for friends' qwalks in the start menu view.



## 4 Domain Model



### 4.1 Class Responsibilities

QwalkGame is responsible for the game flow, initializing and checks with it's quiz which challenges are enabled, and progresses the game accordingly. It communicates with the graphical user interface to make sure the correct amount of markers are visible, when they are supposed to be visible. It also contains the player and monkey (ai) and transfers their information to wherever it needs to be.

AI (monkey/opponent) calculates its own answers depending on the difficulty setting, it also calculates where it should be on the map.

Player only keeps track of its answers to the questions and saves its location.

Quiz has a list of option questions and tiebreakers, a title, a description, keeps track of which settings are on and which are off, stores an id to be identified in the database.

Question and its subclasses has a question, alternatives, the correct answer and a position.





Map gathers user input and also displays markers, the player, the ai, et cetera.

GameTimer helps keeping track of time passed between events. It is used by QwalkGame to measure time taken to finish a qwalk.