

# Report of “Stake Hunting Game” with Mobile phones

made by Arno Lechner & Miriam Fostini

---

## Content

Content .....	1
1 Description and Motivation .....	3
2 Mobile User Analysis .....	3
3 System functionalities .....	4
3.1 The Orientation Game .....	4
3.2 The Questions .....	4
3.3 The Maps .....	4
3.4 The Answers .....	4
3.5 The High score .....	4
3.6 The Upload .....	4
4 Analysis .....	5
4.1 Usage Analysis .....	5
4.2 Screen and Interaction Analysis .....	5
4.3 Environment Analysis .....	6
5 Human Interaction .....	7
6 Architectural Design: .....	10
6.1 Threads: .....	10
6.2 Event Diagramm: .....	11
6.3 Sever Live Cycle: .....	12
7 Code Structure: .....	13
7.1 Mobile Phone: .....	13
7.1.1 MIDLET: .....	13
7.1.2 Data Classes: .....	13

7.1.3	Record Storage .....	15
7.1.4	Filters.....	15
7.2	Server:.....	15
7.2.1	Servlets:.....	15
7.3	Class Diagramms: .....	16
7.3.1	Items Common: .....	16
7.3.2	Filters.....	16
7.3.3	Mobile Phone .....	17
7.3.4	Server .....	18
8	Technical Problems and Solutions .....	18
9	Conclusion .....	19

## 1 Description and Motivation

The Stake hunting Application or also called the Orientation Game is an Game which asks the User different questions and he or she has to solve these. To do that it is necessary to go to this place to get the Answer.

The Main Idea was to have a system which allows downloading from the Server the Game instructions, so that the Game can be played. After the Game is played it should be updated on the Server, so that it is possible to see a statistic on the Web server.

## 2 Mobile User Analysis

Stake hunting or Orientation Game would be an application for children and young adults. The software could be used in Schools to combine new Technology with practical learning and, of course, it could be used to manage Spare Time activities.

This Application tells some facts about the sightseeing attractions of different cities and villages and makes it possible to visit them.

For the navigation the user has access to Google Maps.

Needed are also:

- access to the internet
- download the available Game List with questions and right answers
- access to a map as help option – only with

The Users of the Software will need a phone with Wireless Connection to get the Data from the Server and send the Logs. It will run on J2ME compliant devices.

### 3 System functionalities

#### 3.1 The Orientation Game

The User has to start the application. At the beginning a start Logo will be shown and it will be leaded directly to the main menu. Here he has the possibility to check the list of available games and to download the one he prefers.

#### 3.2 The Questions

The Questions of the Game are downloaded form the server. At the beginning of the Game the questions will be displayed one after the other. The user has the possibility to answer by selecting one of the three choices if it is a multiple-choice question or to input some Text if this is the case.

For each question the player can earn some points. The points will be stored on the Mobile Device.

#### 3.3 The Maps

The User has the possibility to access to some help by opening a map. This will lead to some lose of points.

#### 3.4 The Answers

The Answers the user gives are compared with this ones which are stored on the mobile device. If one answer is right, the points per question are set.

#### 3.5 The High score

The High score contains the names of the completed games and the points the user got. The points are calculated dividing the provided points by the seconds needed and a constant which represents some minus points for the help if it is used.

#### 3.6 The Upload

At the end of the game the user is asked to Upload the result on the server. If this is not possible for some reason or the user wants to do is later on, this is possible by browsing the main menu and selection the item "Done Games". Here the done results can be uploaded every time.

## 4 Analysis

### 4.1 Usage Analysis

- Receive Game data from the Server
- Show Screen with Task/Question and the user should be able to enter the answers per multiple-choice. The Question will be shown in a randomized way, except of the last question.
- The Answer will be stored within the Time and the Location from the Location-API
- After the Location is same as given and the answer is registered the next Question will be shown.
- At the End the user will send the data to the server.

### 4.2 Screen and Interaction Analysis

- The User has to start the application. At the beginning a start Logo will be shown and it will be leaded directly to the main menu.
- To begin a Game the user has to download one form the Server. To do this in the Menu point download all available Games are displayed and the user has to select the one he wants to play.
- After the Download he will be able to start the game. In the same time the start time will be logged.
- After starting the Game the firs Question will be shown on the screen appearing in a From.
- The Answers can be entered by the User on the same screen, depending on the question, multiple-choice or a text field will be shown.
- For each Question Information will be logged like the start and endtime, the points the player got.
- For each Question it is also possible to access GoogleMaps over the Internet. A map of the area in which the cell phone is detected by the Location API is displayed.
- At every time it will be possible to exit the game without stopping it.
- At the end the user is asked to Upload his game results. If he does not he will be asked later.
- The done game is moved to the "Done Games" Menu point.
- The points reached are shown in the High score.

### 4.3 Environment Analysis

- The server connection will be just used to upload and download data from the server.
- We will use the post method over HTTP Technology.
- The Data will be transported via W-LAN to the Server over the Internet.

## 5 Human Interaction

When starting the user has 2 options:

- Download: Downloading a game form the Server
- Play: Playing a Game which has been downloaded before
- Rules: Read the rules of the Game
- High score: View the High score if there are already some entries



The menu:

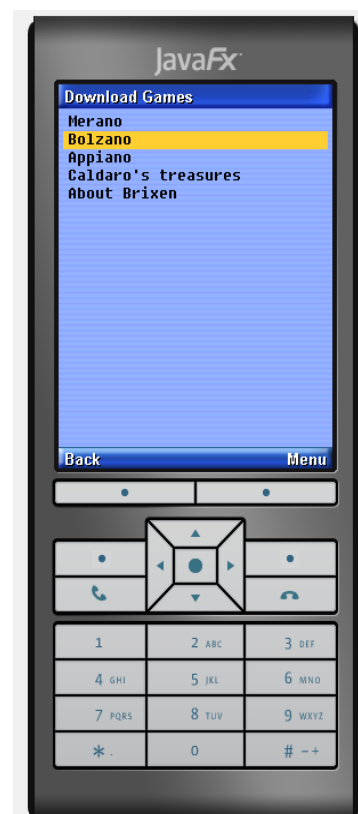
Download leads to the list of available games.

Done Games contains the Games which are already terminates.

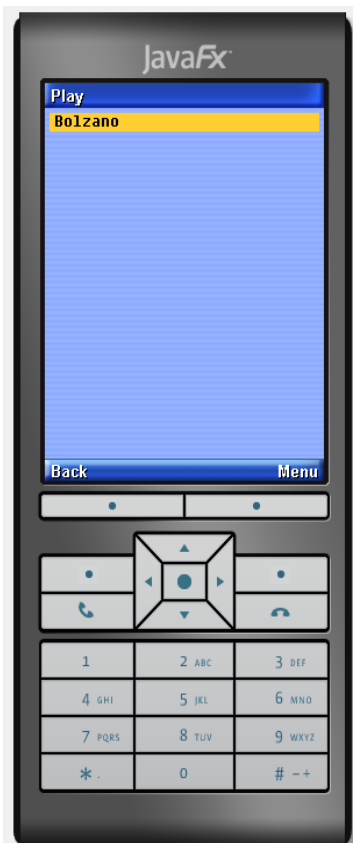
Rules will display the rules of the Game.

High score contains the points reached per game.

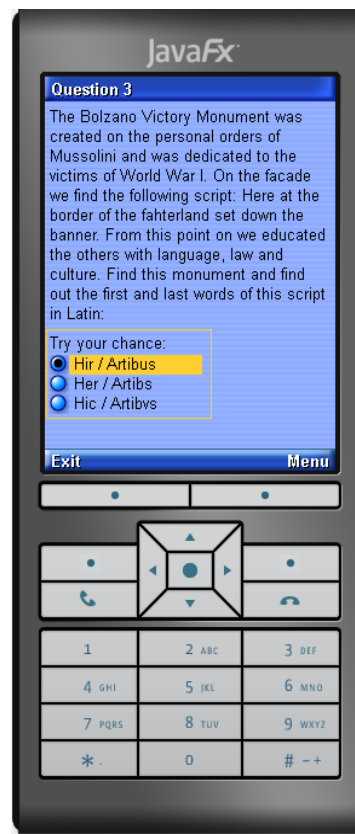
Info contains the user Id used on the server.



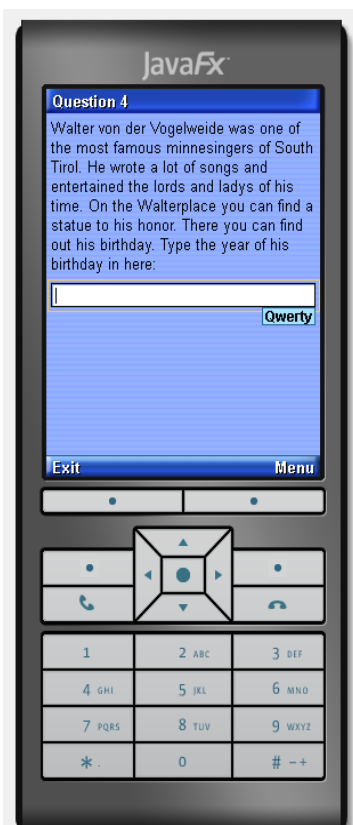
Choose one of the available Games and Download it...



The downloaded Game is been transferred in the "Play" directory. There it can be started...



The Questions are displayed one after the other. For the multiple- choice ones the player has to choose the right answer. Each Question tells some facts.



If there is a Textbox, the user has to enter the right answer by itself...



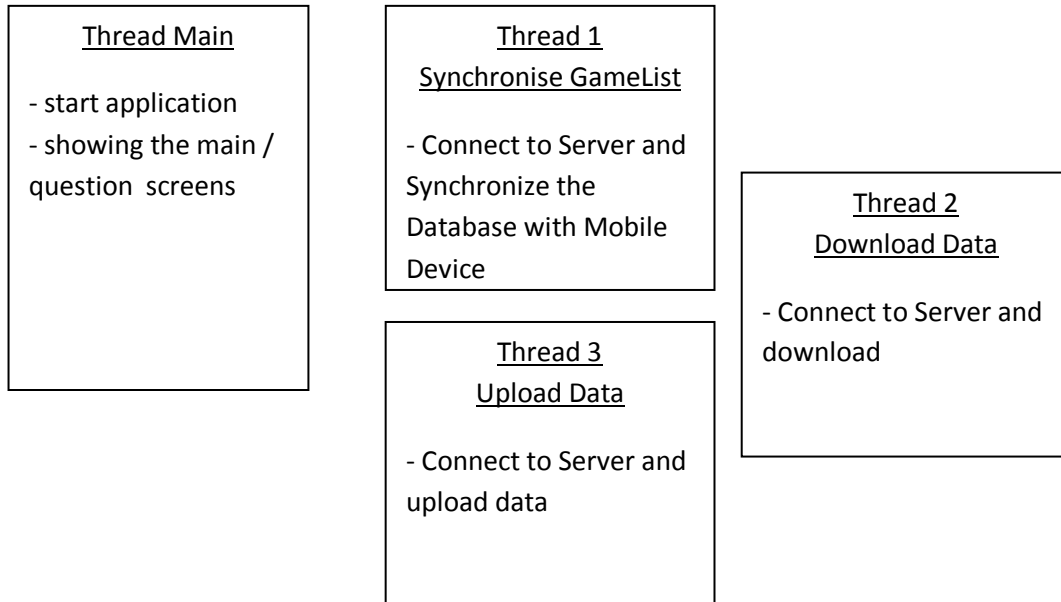
For each Question in the Help menu can be found a map of the actual location.



At the end of each Game the the game and its results are Uploaded to the Server.

## 6 Architectural Design:

### 6.1 Threads:



This Diagram shows the Threads which are in our phone implemented.

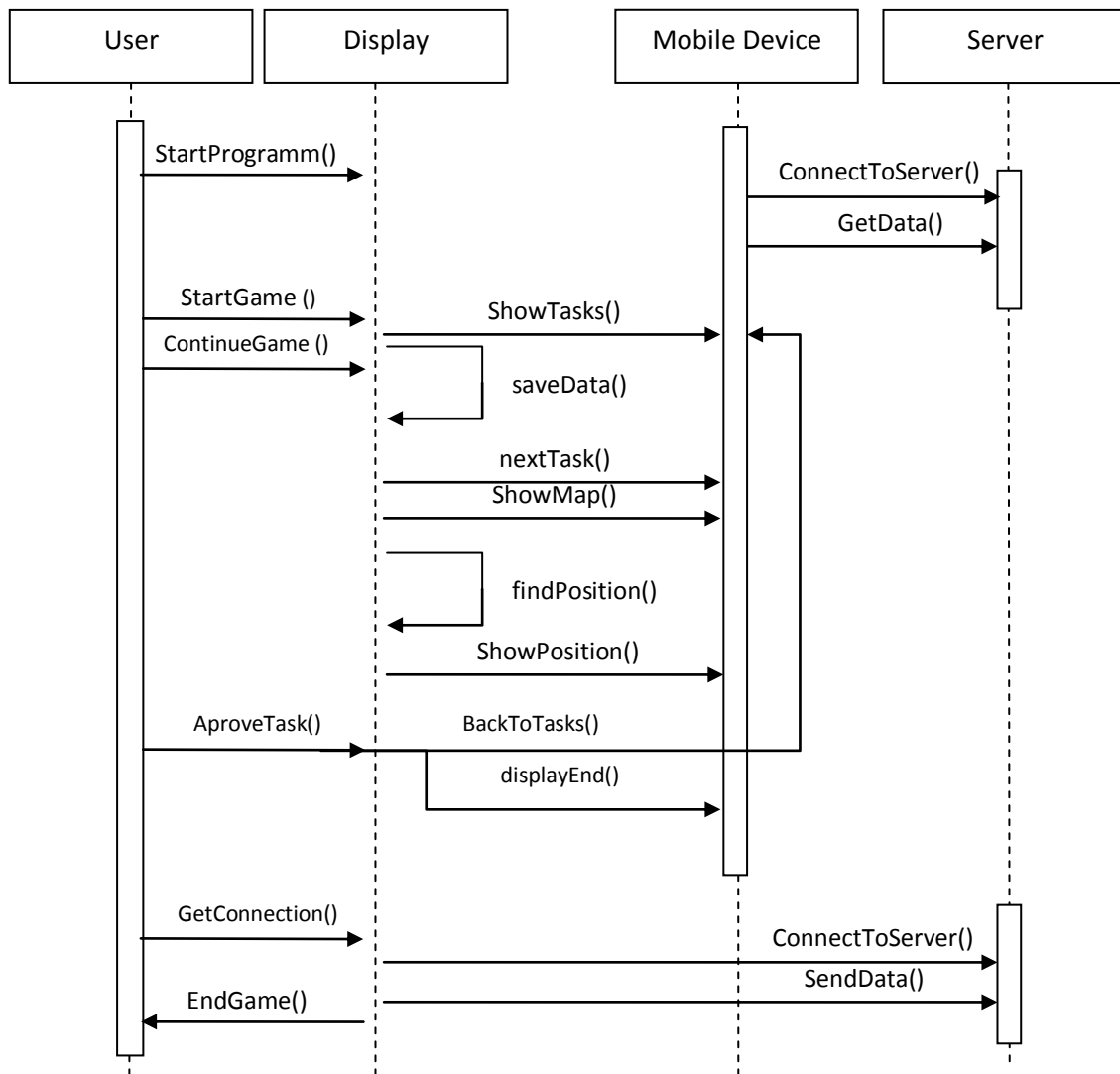
The Main Thread is the main program which calls the 3 over Threads.

The First Thread is been used to synchronies the Game List on the Mobile phone with the Game List on the Server. It sends the data from the phone to the server and receives the Full Game List.

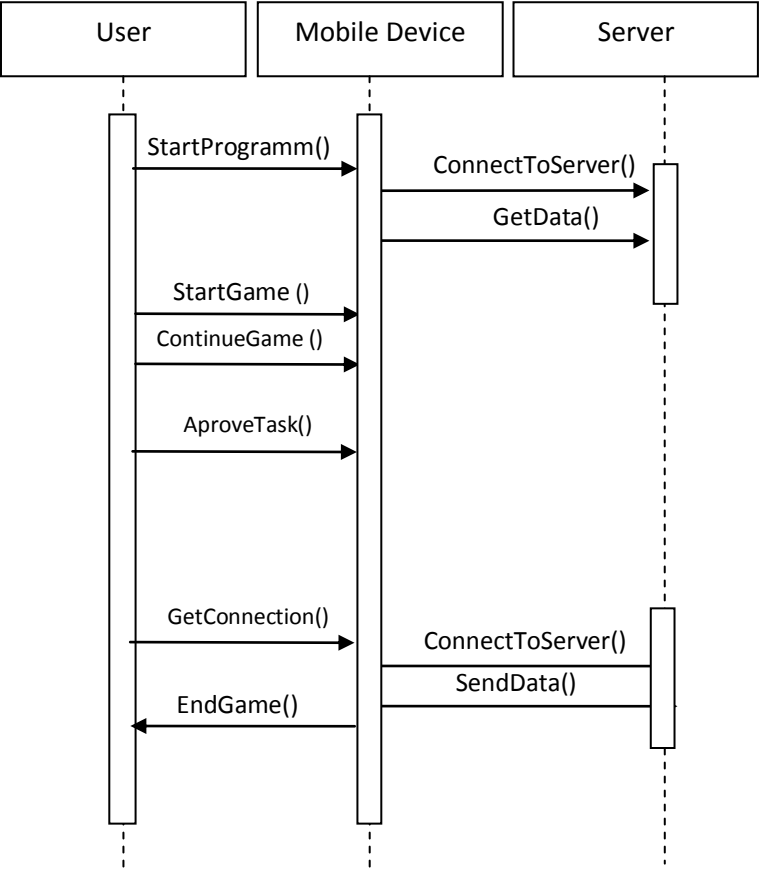
Thread 2 is responsible for the Downloading the selected Game from the Game List. It just sends an id to the Server and receives the Questions and Answers.

Thread 3 has the task to upload one Game to the server it does not receive any data.

### 6.2 Event Diagramm:



6.3 Sever Live Cycle:



## 7 Code Structure:

### 7.1 Mobile Phone:

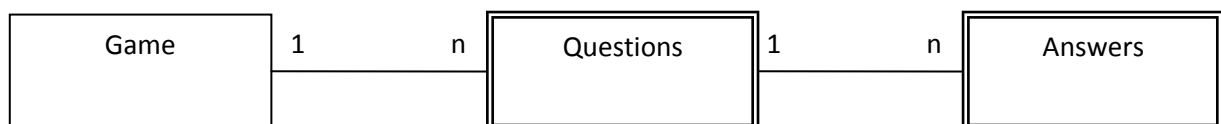
#### 7.1.1 MIDLET:

The MIDLET is responsible for the initialize the Application on the Phone. Which is also Calling the Method of Main Menu List which will then been shown on the Screen. It also has the methods to set the Displayable from the over classes. This MIDLET will be passed through the whole program, so that the forms and alerts can be shown. Also the actually Game will be called on that Class.

#### 7.1.2 Data Classes:

There are 3 Data Classes, which are storing the Information on the Object. These also have implements methods to read and to write on an DataStream.

Entities:



##### 7.1.2.1 GameItem

This Class Stores all information about the Game.

Field Name	Type	Int. Value	From DB / User
Id	Int	0	DB
Name	String	""	DB
Info	String	""	DB
Location	String	""	DB
Date	String	""	DB
pointsAll	Int	0	User
startTime	Int	0	User
endTime	Int	0	User

**7.1.2.2 QuestionItem**

This Class Stores all information about the Question.

Field Name	Type	Int. Value	From DB / User
IdQuestion	Int	0	DB
idGame	Int	0	DB
Text	String	""	DB
answerType	String	""	DB
Location	String	""	User
pointsToGet	Int	0	DB
PointsHelp	Int	0	DB
helpNeeded	Int	0	DB
pointsGot	Int	0	DB
startTime	Int	0	User
endTime	Int	0	User

**7.1.2.3 AnswerItem**

This Class Stores all information about the Answers.

Field Name	Type	Int. Value	From DB / User
IdQuestion	Int	0	DB
idGame	Int	0	DB
idAnswer	Int	0	DB
text	String	""	DB
selceted	Boolean	false	User
answerText	Int	0	User

### 7.1.3 Record Storage

There are 4 Record Storages which are been used to save data.

The Data\_Store is for the storing of every data which is not been involved of the Game structure. As for example: the User Id. This Id is more been used to identify the Phone on the Server.

#### 7.1.3.1 *Game\_Store, QuestionStore, AnswerStore*

These storages have the same structure. For Each record is an Object with the information.

### 7.1.4 Filters

The Filters are been used to get the data from the Storages.

## 7.2 Server:

The server is there to connect to the DB-Server and to synchronies the Game List with the DB.

### 7.2.1 Servlets:

#### 7.2.1.1 *NewGameListServlet*

This SERVLET is checking the Game List and send the data back.

#### 7.2.1.2 *DownloadGameServlet*

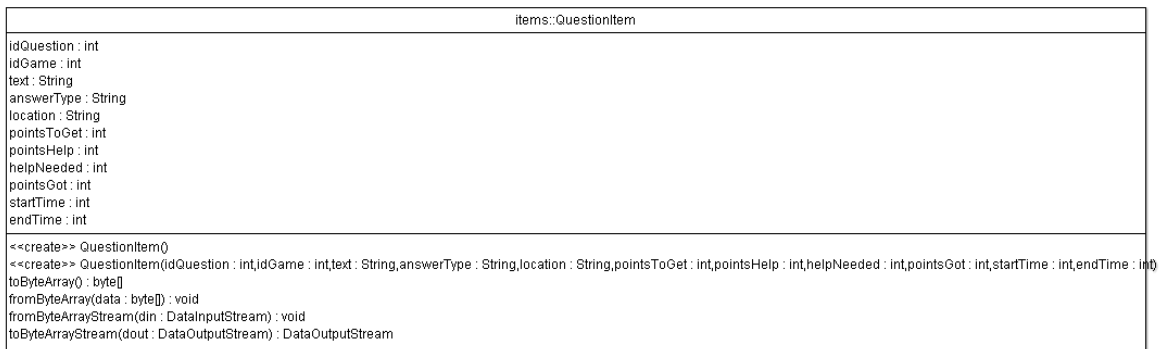
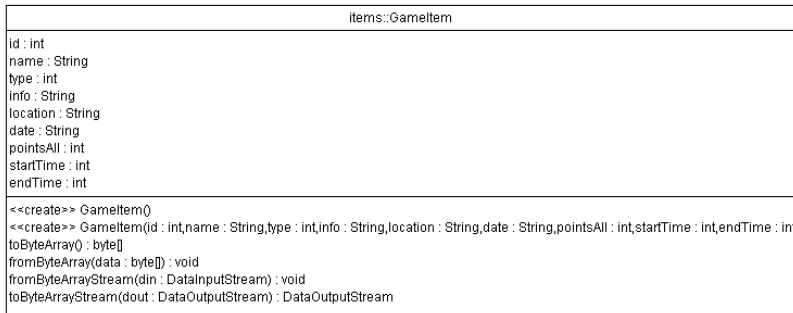
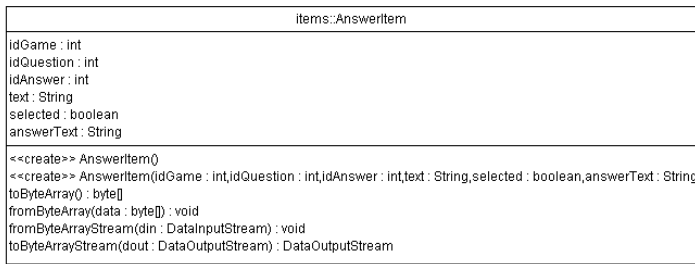
This SERVLET is getting the Information of the Question and Answer from the DB-Server, for the given ID of the Game.

#### 7.2.1.3 *UploadGameServelet*

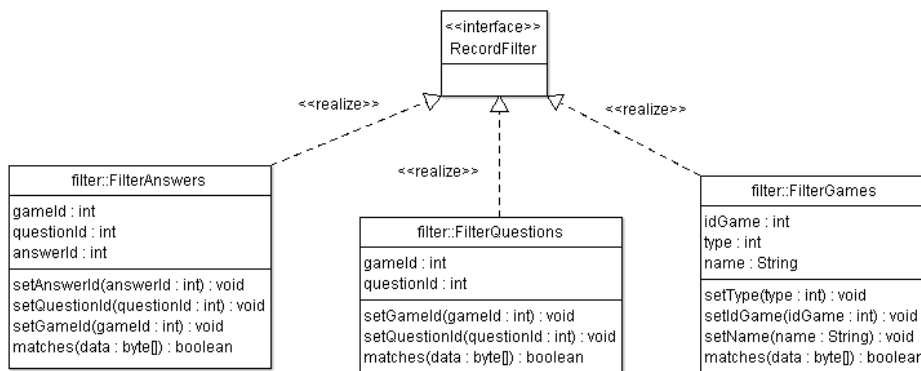
This SERLVET is to upload the Data which the user has modified. These data will be saved in different Tables on the DB-Server with the USER id, so that the data can be processed for each User.

### 7.3 Class Diagramms:

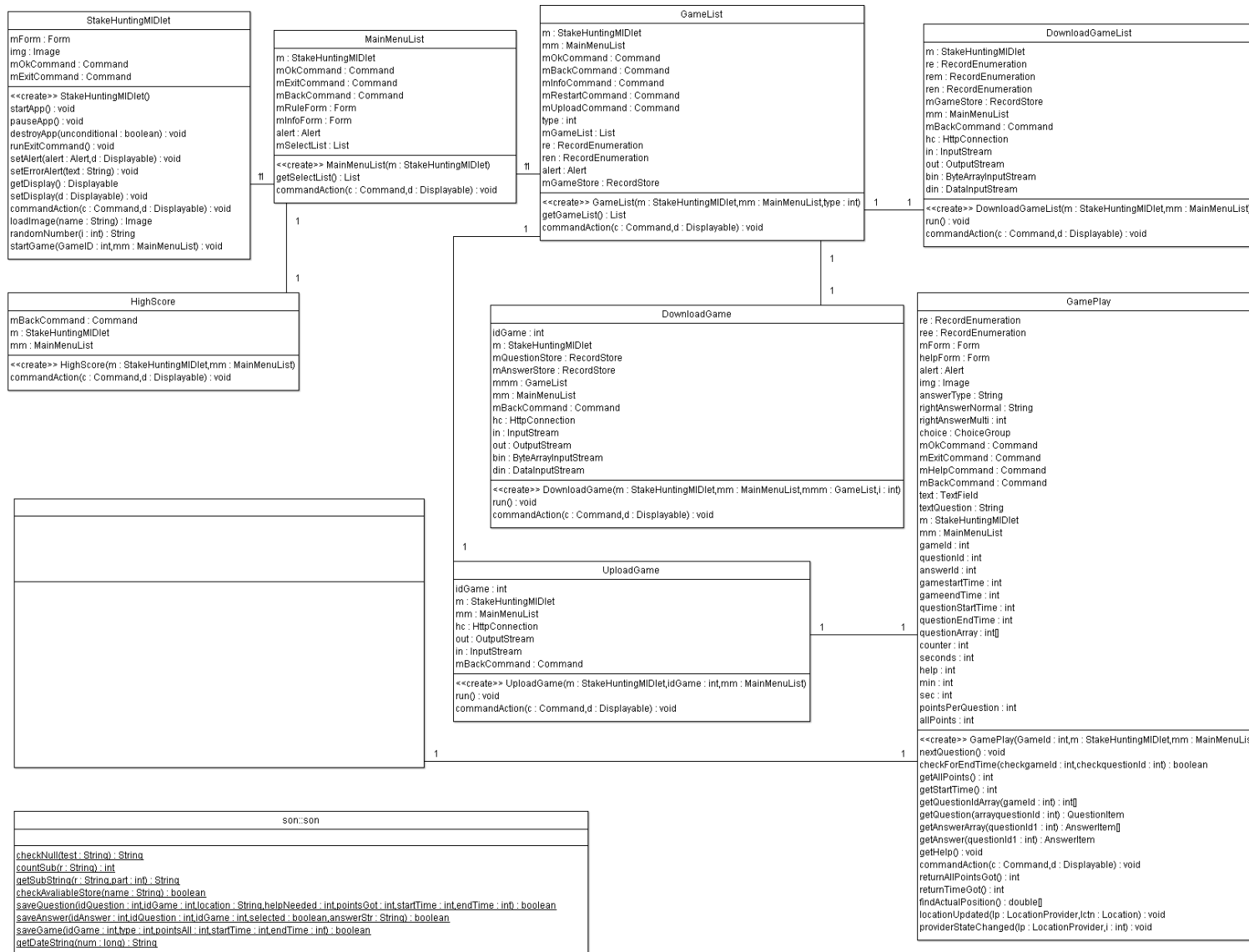
#### 7.3.1 Items Common:



#### 7.3.2 Filters



7.3.3 Mobile Phone



### 7.3.4 Server

mobile::DownloadGameServlet
dbAnswer : ArrayList
processRequest(request : HttpServletRequest,response : HttpServletResponse) : void doPost(request : HttpServletRequest,response : HttpServletResponse) : void getDataQuestions(con : Connection,idGame : int) : ArrayList getDataAnswers(con : Connection,idGame : int,idQuestion : int) : void

mobile::NewGameListServlet
processRequest(request : HttpServletRequest,response : HttpServletResponse) : void doPost(request : HttpServletRequest,response : HttpServletResponse) : void checkDb(con : Connection,din : DataInputStream) : String synchroniseList(dbList : ArrayList,mbList : ArrayList,con : Connection,num : int) : ArrayList getGameById(con : Connection,id : int) : GameItem userPresent(con : Connection,idUser : int) : boolean userInsert(con : Connection,idUser : int) : void

mobile::UploadGameServlet
processRequest(request : HttpServletRequest,response : HttpServletResponse) : void doPost(request : HttpServletRequest,response : HttpServletResponse) : void saveGameToDB(con : Connection,idUser : int,game : GameItem) : void saveQuestionsToDB(con : Connection,idUser : int,questions : ArrayList) : void saveAnswersToDB(con : Connection,idUser : int,answers : ArrayList) : void

son::son
checkNull(test : String) : String getSubString(r : String,part : int) : String checkNullB(test : byte[] ) : byte[]

## 8 Technical Problems and Solutions

The main challenge was to create an ideal user interface and to implement the Download and Upload in a simple and effective way. Therefore we implemented a Synchronization method, with which it is possible to check the Database and the Record Store on the mobile phone. Through the synchronization it is possible to download only the relevant things such that on the mobile phone are stored only the things really needed in order to not waste the precious storage space.

Another challenge was to implement the Help Menu in terms of a map. First we wanted to implement the whole map with the Location API of the Mobile Phone, but we find a better solution in Google Maps. The disadvantage is that the user has to have a continuous internet connection to see the map, but implementation is much simpler and the map has a nice and user-friendly look. We used the Location API to get the actual position of the Mobile Phone. The Longitude and Latitude are returned and passed as parameters to the Google Map method. In this way each time the Help Menu is accessed the area where the user currently is will be shown.

## 9 Conclusion

During the Development of this Project we learned a lot about the theory of mobile software, and also we gather a whole new experience to do it better in the Future. We see this project more like a Prototype an how the Idea could be implemented.

For the next Time we would to choose more likely a Web service, which works over XML Files. Over HTTP we had problems with the Encoding and Decoding of the Strings.