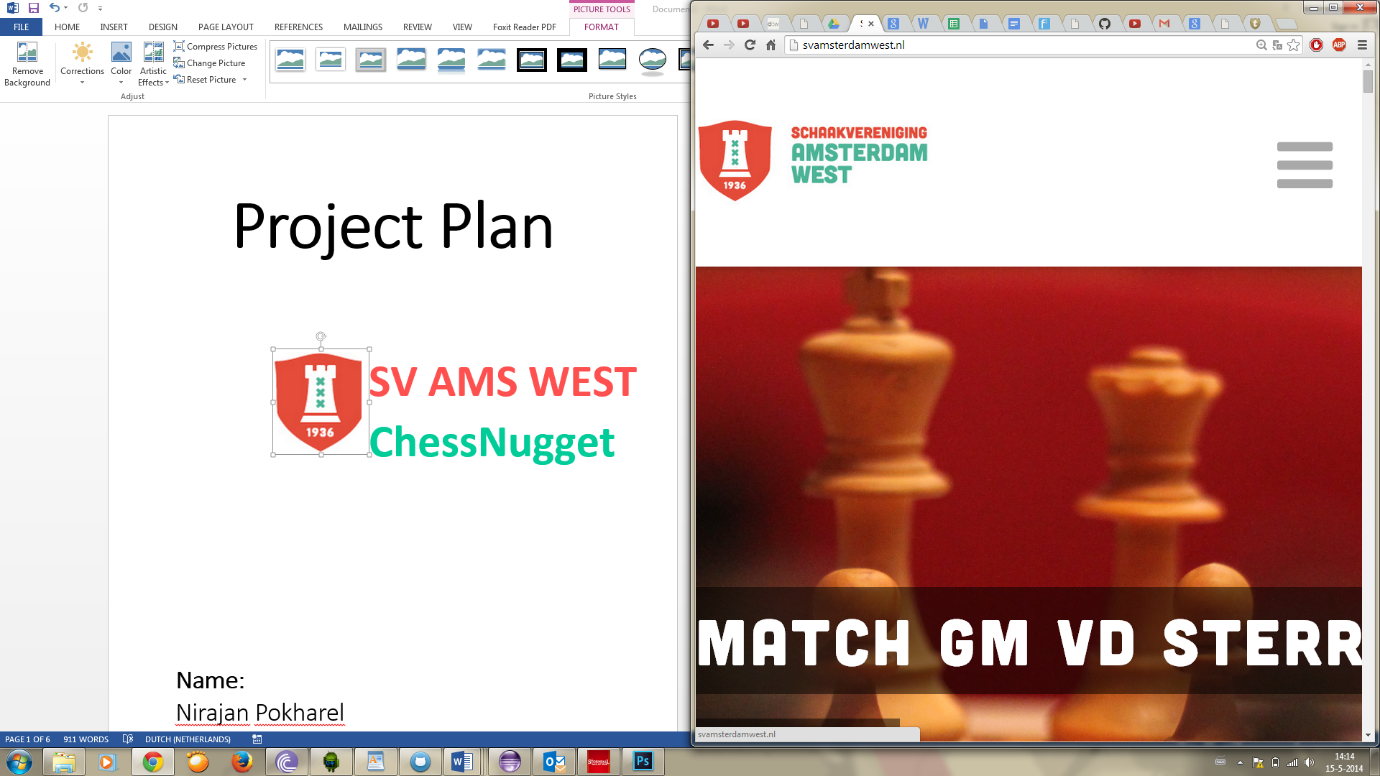
**SV AMS WEST   
ChessNugget**



**Name:**   
Nirajan Pokharel  
Kinson Michel  
Daniel Jeantihomme

Tobias Winkels-Herding

Project Android Development

Preface

As part of Inholland University of Applied Sciences School, IT Mathematics and application course curriculum, in the fourth semester in the third year the students had android an development project. This project was based on the Android development class with combination of statistics and algorithms.

The project consists of design and programming an actual app, but also finding an client to develop this app. the lecture task few base functionalities which the client add to give suggestion into in improving it and making more useful.

This presented an opportunity for the student to connect the dots of what was thought during these courses and also enabled the student to prove thier skills, improve and learn much more.

Special appreciation going out to …… and ocscar for their patients and guidance. These people have invested time and sweat in guidance and help with this project and our personal development.

**Table of Contents**

\*dummy text

[Introduction | Chapter 1 3](#_Toc384484073)  
[PROJECT AFFAIRS | 1.1 4](#_Toc384484074)  
[Team Agreement | 1.2 4](#_Toc384484075)  
[RESEARCH QUESTION | 1.4 4](#_Toc384484076)  
[METHODOLOGIES | 1.5 5](#_Toc384484077)  
[PLANNING | 1.6 6](#_Toc384484078)  
[Research |2 3](#_Toc384484073)  
[Developement Method and Requirements specification | 2.1 4](#_Toc384484074)  
[Look and Feel | 2.2 4](#_Toc384484075)  
[Testing | 2.3 4](#_Toc384484076)  
[Test Results |2.4 5](#_Toc384484077)  
[The final product “Chessnuggets” |3 6](#_Toc384484078)  
[System architecture | 3.1 3](#_Toc384484073)  
[Database design | 3.2 4](#_Toc384484074)  
[ENTITIES Client side|Section 1.1 4](#_Toc384484075)  
[Relationships |section 1.2 4](#_Toc384484076)  
[communication| 3.3 5](#_Toc384484077)  
[Application| 3.3 6](#_Toc384484078)  
[Win/lose Statistic Algorithm| 3.3 4](#_Toc384484074)  
[Pairing Algorithm| 3.3 4](#_Toc384484075)  
[Conclusion and recommendation| 3.3 4](#_Toc384484076)  
[PLANNING 5](#_Toc384484077)  
[REFERENCES 6](#_Toc384484078)  
[METHODOLOGIES 4](#_Toc384484076)  
[PLANNING 5](#_Toc384484077)  
[REFERENCES 6](#_Toc384484078)  
[INTRODUCTION 4](#_Toc384484074)  
[RESEARCH QUESTION 4](#_Toc384484075)  
[METHODOLOGIES 4](#_Toc384484076)  
[PLANNING 5](#_Toc384484077)  
[REFERENCES 6](#_Toc384484078)

Summary

The client for this assignment is “schaakvergeling Amsterdam west”.

So far it the one of the youngest association in the Netherlands. Amsterdam West Chess Club, it was found on June 27, 2013 and merge with “TAL/DCG” and “The Problem”. The end result was a brand new association. A brief history on the two merging partners. The Problem was found on April 21, 1936 and “TAL/DCG” was found in the 1936 as well. This suggest that both club have been establish for about 78 years.

As part of the community the Amsterdam west, the chess club have been focusing on the youths. Therefore when the development team contact them and explained the opportunity to get a free android app design and implemented, they were more than welling. The project is focused on making an app that will help player during a tournament with tips to improve their chances of winning.

# Introduction | Chapter 1

This chapter will describe the project; the project plan, the problem and the purpose.

Preface &original Task

Chess is one of the biggest indoor sports in the world, with many tournaments held locally and internationally each year and players of all skill levels participating (Peter (2012)).

During the chess tournaments player like to be prepared, by analyzing their opponents style of play, such as their preferred openings, evaded openings and lategame strengths and weaknesses.

The information gathered from their analysis helps player to strategize against their most likely opponent Harald (2014). This method can only be useful, if one has an idea of who their next opponent will or might be.

In this project an Application Working Title “ChessNuggets” will be build which is based on the android platform. The app will be use by any chess player participating in a chess tournament.

This app will gather information on players in order to predict their most likely future opponent, possible weakness and strengths.

**PROJECT AFFAIRS | 1.1**

Project team: Roles:

Nirajan Pokharel Scribe, Glossary keeper, Guru

Kinson Michel Guru, Project manager

Daniel Jeantihomme Time Keeper, Guru

Tobias Winkels-Herding Guru, Client interaction

Team Meeting schedule

|  |  |  |
| --- | --- | --- |
| Date | Time & place | Comment |
| Thrusday week1 | School x2-10 | Work 3 hours |
| Wednesday |  |  |

**Team Agreement | 1.2**

# Assignment assigned to team members should be delivered at scheduled date. If one cannot meet this date, the project manager should be consulted as soon as possible to make a work around.

# Slacking prevention -One is allowed to be absent once -Second absent one receive a warming -Third absent strong recommendation to leave group

# RESEARCH QUESTION | 1.4

How can an android app be created, that can gather data on players and process them into statistical information such as odd of winning in respect to chest tournament?

* [1] What kind of architecture can be used to build the ChessNuggets app?
* [2] How can the GUI be design?
* [3] How can the rating system and tournament historical data be gathered and implemented?
* [4] How can the app be tested?

# METHODOLOGIES | 1.5

The methodologies described below are based on the primary and secondary research question.

For research question [1], a detailed literature study and a proof of concept about relevant architectural software design will be conducted to build the best architecture design for ChessNuggets app.

From this literature study, it is expected to find the best architecture used for building an android app.

At this point there should already be an overview over performance, type database and classes that need to be design.

[2] The GUI for the ChessNuggets will be design based on detailed interview with the chess club client representative.

A literature study will also be conducted for the guideline need for the GUI. There will be a proof of concept (POC) that will be a prototype of the ChessNuggets app.

From this interview it is expected to gain an overview over the color scheme for the app, fonts, touch and feel of the app, usability, and sketch a from the client.From the literature study, it is expected to discover best practice, “dos and don’ts “for the app design.

The idea is to make the app dynamic in order to be used by any android device. The app will provide an optimized layout for lower memory usage (http://developer.android.com/training/improving-layouts/index.html ).

The POC will use action bars design instead of menu buttons in order to give a top level user experience. After the completion of the POC the client is to be consulted wether the prototype is up too expectation. If not it is still possible to implement changes at this early stage of development.

For research question [3], a literature study on chess rating system will be conducted wich will lead to a reliable source regarding chess player ratings information to be implementing in the app.

Since realiability and accuracy is important the second client interview will focus (next to the interface design) on specific algorithms used in the chessclub, derivations from known algorithms and other problems that might occur with the algorithm.

For research question [4], again a literature study will conducted. This time about relevant testing methods for android app.

From this study, it is expected to find the best possible method for testing Android applications.

# PLANNING | 1.6

Below is a table with the planning, it describes what the planning for each week. The table is separated into two parts; the first show the week and the topic of the week, and second describe the topic in further details.

|  |  |
| --- | --- |
| WEEK | PLANNING |
| Week 1 research | -Everyone will conduct research on their assigned research question  -The result acquired from the research will guide one into implementing phase of the project |
| Week 2 Basic Implementing | Implementing basic methods & classes |
| Week 3 Implementation GUI | Implementing The user interface |
| Week 4 Implementation Algorithm | Creating an algorithm |
| Week 5 Implementation GUI & Algorithm | Using data (rating, style of play & etc.) with the implemented code |
| Week 6 Testing, documentation & report | Testing for bug, documenting the work and writing the report |
| Week 7 report | Finalizing the report |

### The goal of this Android App is to aid tournament participant during a tournament. Participant with this app installed can inform the club in advance weather they will be attending or not. Tournament participant can see their opponents in advance and will statistic data give them an idea who will have the upper hand during that match. To explain this structure, this assignment has been broken down into several chapters:

* Chapter 1: This chapter gives a more detailed explanation of the assignment
* Chapter 2: This chapter will describe the research findings and testing pahse
* Chapter 3: This chapter describe the final product and it specficiation, conclusion and recommendation

.

# Research |2

This chapter will give more information regarding the assignment’s methodology.

**Developement Method and Requirements specification | 2.1**

The development method “MoSCoW”[ Tierstein, L.M. (1997)], providing flexibility and scalability, was used in order to Implement this project.The Group and the client together came up with the following Requirements for the application.

**MUST**

**statistical information**This information will give the user an idea of possible outcomes of a match against any player, so weather they have a high chance to lose, win or draw a match.

**Pairing algorithm**This algorithm will display the user their most likely opponent.

**Competition Overview**This algorithm will display the user their most likely opponent.

**Internet connection**In order to keep the information up to date, frequent resyncing is required. In order to resync an internet connection is required, therefore the app will inform the user about the availabilty of internet.

**Player registration/present**Player can register into the app allowing them to retrieve personalised data. Once registered the player can inform the club about his attendance during the next meeting inside the app. This information is stored on a server and availeable to all other users of the app at any given time.

**Translation from Database to Database**The client uses a homebrewed Database structure, in order to minimize redundant Data this Database structure is imported into a mysql Database and normalized in the process. Afterwards it is exported to the Application via JSON strings.

**SHOULD**

**News**

The App Displays the “News section” that is also availeable on the clubs homepage.

**Historical** **Data**

The App stores Data over several Seasons and makes predictions and statistics based on it.

**Admin allowed to sign in and out other players**  
Admin users can sign users in and out of the next meeting, same as Users can do for themselves.

**COULD**

**Calendar**Club events could be displayed in an in-app calendar.  
  
**Change User password**Admins can change the passwords of users to lock them out of the Database or reset their account.  
  
**Support for other Tournament types**  
The Club also plays the tournament types “Kroongroup” and “Beker” next to the internal competition. The app could also implement functionality for these events.

**WONT**  
**Multiple Plattforms**  
Due to no experience for cross-plattform portation this Idea will not be implemented.

**Recording of Games**  
Whilst beeing increadible interesting this idea is too far of the given task and to complex for the scope of this project.

### Look and Feel | 2.2

The majority of the memmbers of the “[Schaakvereniging Amsterdam West](http://www.google.nl/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCUQFjAA&url=http%3A%2F%2Fsvamsterdamwest.nl%2F&ei=BQOhU7fzHcz50gWwuoBg&usg=AFQjCNHBtlCrxFwmJ_XhUPCtegxYPTjWDg&sig2=wfnjlQVnWTMy-cnMqN399A&bvm=bv.69137298,d.d2k)” are older than 50 years [rob Scheermeijer 2014 ].

Seniors are one of the fastest growing demographics to embrace technology although their age may affect their ability to execute tasks [LEE RAINIE 2012]

Developers therefore should keep in mind the color scheme, size of text and the app complexity in respect to this demographic.

For this the “simplify my life” approach was taken. The idea is that the app has to give the user the feeling that “this app makes my life easier and it easy to understand ”.

This means new Users approaching the app for the first time should intuitively graps the most important feature.Simple tasks will not require complex procedures.

The app color scheme was chosen to match the online website/logo of the chess club whilst not hindering members with impaired vision or other eyesight issues.

The screens grew from several brainstorming sessions in the group. See appendix C for a history from the earliest designs to the final product.

Use case diagram|2.2.1  
(see appendix B)

### Testing | 2.3

During the development process the app was constantly beeing tested to avoid redundancies and bugs.

As a primary test the app will be installed on all group members mobile phones in order to test core functionality.   
The secondary test will simulate several rounds of tournament play by uploading several of the clients database files to the server whilst emulating club members on the groups telephones in between.   
  
Tertiery testing will commence inSeptember after the Summer break, when the next competition is starting. The app will then be distributed to members of the club and tested “in action” for several weeks to solve the last remaining bugs and end the development cycle.

Test Results |2.4

Result from the primary test suggest that the app requires several minutes to import all the data after a fresh installation, this might be frustrating to some users. The rest of the core functionality of the app however worked with little error.

The Secondary testing was supposed to happen after receiving multiple files from the clients datastructure.

# However the client switched Databases during development leaving the development team with only one Database file to test and develop. Since the “[SchaakverenigingAmsterdam West](http://www.google.nl/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCUQFjAA&url=http%3A%2F%2Fsvamsterdamwest.nl%2F&ei=BQOhU7fzHcz50gWwuoBg&usg=AFQjCNHBtlCrxFwmJ_XhUPCtegxYPTjWDg&sig2=wfnjlQVnWTMy-cnMqN399A&bvm=bv.69137298,d.d2k)” is also taking a summerbreak no newer files are beeing produced. Secondary testing was therefore postponed until more Database files can be aquired.

# The final product “Chessnuggets” |3

Chessnuggets is the finished Android Application that was built to help players and planners of the “[Schaakvereniging Amsterdam West](http://www.google.nl/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCUQFjAA&url=http%3A%2F%2Fsvamsterdamwest.nl%2F&ei=BQOhU7fzHcz50gWwuoBg&usg=AFQjCNHBtlCrxFwmJ_XhUPCtegxYPTjWDg&sig2=wfnjlQVnWTMy-cnMqN399A&bvm=bv.69137298,d.d2k)” with the internal competition of the Chessclub.

### Final implementations |3.1

Users can perform the following tasks in the application;

Check chance of winning

View Possible opponent

View statistics about self and other club members

Sign in and out of the next club meeting

Additionally Admins can perform the following task in the application

Sign Users in and out of the next club meeting

### Language | 3.2

For the Front end ( client) the Android application was programmed using java [] in the eclipse environment. The back end (server,) was written with PHP and JavaScript due to familarities and ease of migration to the clients server.

### -Architecture -DB Design -Entities -Communication -Server side -Client Side -Algorithm

### System architecture | 3.1

This app is installed on multiple devices that are supposed to communicate with each other (share coming/not coming). As a direct consequence of that a central point of communication must be established. For this reason a client-server model architecture was chosen. The device is the client side of the architecture. The client side gather information about availeabily of each player. This information is then fed to the server and distributed back to all clients.

### Database design | 3.2

During one of the meetings with the project supervisors the group came up with a database.

For simplicity a similar model was designed and implemented on the client and server side. The server has the same design as the client barring some additional tables, and some fewer columns in the player\_chessapp table.

This is due to the fact that some Data is not needed on the client side for the first part and due to the fact that android limits queries for the second.

   ENTITIES Client side|Section 1.1

All entities were created based on the initial design developed during aforementioned meeting. The entities that were created are PLAYERS\_CHESSAP, MATCHES\_BEKER, MATCHES\_INTERNAL, AND MATCHES\_KROONGROUP and MATCHES\_ABSENT.  They are as follows:   
  
1. Player table  
  
 In the entity players\_chessapp, the Identifier is the ID. ID was chosen because no player has the same player number, therefore player number is unique.

# 

ID#

fname,   
lname,  
 next\_pairing,  
 points, current\_rating, rating\_startofeason, currvalue,

players\_chessapp

2. MATCHES\_BEKER, MATCHES\_INTERNAL, AND MATCHES\_KROONGROUP  
In the entity matches\_beker, matches\_internal, and matches\_kroongroup the Identifier is round and id\_player\_white. round and id\_player\_white was chosen based on the procedures of normalization of the database.

round, id\_player\_white,  
  
 id-player\_black, winner\_id

matches\_internal

round, id\_player\_white,  
  
 id-player\_black, winner\_id

matches\_kroongroup

round, id\_player\_white,  
  
 id-player\_black, winner\_id

matches\_beker

3. Matches absent  
In the entity Matches\_absent, the Identifier is the round and player\_id.

round,   
player\_id,  
  
 code,   
timestamp

Matches\_absent

Relationships |section 1.2

Player\_chessapp

matches\_beker

Player\_chessapp

matches\_internal

Player\_chessapp

matches\_Kroongroup

The chess club structure suggest that a player can have multiple Tournaments so therefore the MAX cardinalities was chosen after reviewing the chess club tournament, another conclusion was each player can participate in at least one match therefore MIN cardinalities was chosen. A match can be many players so therefore this was chosen as the MAX cardinalities and a matches must have atleast two players so the MIN (optional) cardinalities was derived.

Player\_chessapp

matches\_absent

In Figure 1.1 one can see a clear picture of the relationship of the entities

Employee#

EmployeeFirstName

EmployeeLastName  
Function  
Start Date  
Salary  
Password

Player

round, id\_player\_white

id-player\_black, winner\_id

Match\_internal

round,   
player\_id,  
  
 code,   
timestamp

Matches\_absent

round, id\_player\_white

id-player\_black, winner\_id

Match\_beker

round, id\_player\_white

id-player\_black, winner\_id

Match\_Kroongroup

Figure 1.1

### communication| 3.3

The server and client communicate through use of http protocols. They utilize httpRequest and HttpReply. The httpRequest is sent from the client side to the server. These requests are sent using GET method, however for security reasons all non publicly availeable Data requires a key that only club members can obtain.  
  
When the httprequest is received on the server side one of the following php files are called; change\_absent\_admin.php, change\_absent\_personal.php, upd\_player\_matches.php and verify.php. The in-depth functionalities of these files can be found in thechnical server file documentation.(see appendix A for more information)  
  
The information that is sent back to the client are sent in JSON format. JSON strings make communication and implementation of the information very easy in any language because of their lightweight, minimalistic and highly portable structure that relies only on two fundamental things; a collection of name/value pairs, and an ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

### Application| 3.3

 When a user start the app it is required to sync with the database first. The app was not programmed to auto sync to give the user more control over their Data usage when not on a Wireless network.  
When updating from the server the app will provide a timestamp of the most recent updates. This time stamp is used to determine which data has been changed since the last update. Only information that was changed will be delivered to the client again making the app sync more efficient and less time consuming. On update the Json string is received with the changed data (see sever side for more detail). This string will contain:

If changed since last update:  
 All changes to player showing up in the next meeting.  
 If changed since last update:  
 All changes to rating per player  
 If changed since last update:  
 All wins/losses per player  
 (Will be implemented later)  
                               if updated since last update:

**News.**  
Because of time constraints the news will not be implemented in this version but will be implemented in upcoming version.

The idea is the news will be generated from the database on the WordPress page that is used by the client. The news from the web application contains a menu filter, which allows to show new from certain segments. Calendar is also another function that wasn’t implemented because of time constraints.

**MAINMENU**  
On the main menu there are five option. Thses option as follow:

FAQ  
Frequently Asked Qusetion is page contain step how to operate the app and also answers to question the user might have.

Tournament  
The Tournament will send the user to a new activity.  If the user has no tournament it will inform the user that he needs to be in a tournament. Then the app will show the user his tournament. The tournaments use the Swiss style. The user will see a three tabs

Likely opponents: this tab this plays the user top 3 likely opponents.  
Player list1: This tab displays a players list sorted ranking base on rating  
Player list2: This tab displays a players list sorted ranking base on points

**Sync data**  
Sync data simple contacts the server use httpRequset.

**Profile**  
the player information is available. The player can see their name and ranking. There is an option indicate if you’re attending or not. There is also a chart that the user can click and display their progress so far in the tournament.

**News**

                As mention previous the news will be implemented in later versions

### Win/lose Statistic Algorithm| 3.3

 Use the elo rating system and their rating chance on winning was calculated. The elo is a rating system is used to calculate skills level of one’s competitor. An ELO rating system is a system used in chess to calculate an estimate of the strength of the player, based on his or her performance versus other players. Most of the systems are used to recalculate ratings after a tournament or match but some are used to recalculate ratings after individual games. In almost all systems a higher number indicates a stronger player. In general, players' ratings go up if they perform better than expected and down if they perform worse than expected. The magnitude of the change depends on the rating of their opponents. The Elo rating system is currently the most widely-used.

Below is the formula to calculate possibility of winning base on a players rating

E_A = \frac 1 {1 + 10^{(R_B - R_A)/400}}.

Wikipedia 2014- elo ranking

### Pairing Algorithm| 3.3

 The goal of this Android App is to aid tournament participant during a tournament. Participant with this app installed can inform the club in advance weather they will be attending or not. Tournament participant can see their opponents in advance and will statistic data give them an idea who will have the upper hand during that match.{ With the help of this app player knows about their best pairing given by percentage as chances to play the next match. The first probable player have percentage 90, second 80 and the third with 70 percentage. All those percentage are calculated based on player previous match history. First thing that is checked in-order to calculate the probability is if the player are playing External and or are absent, also if the pair played beker(tournament)and krounegroup(tournament) for 5 times the condition will be that they cannot be paired. There is also a condition in which if the player are paired manually by the admin then that player is the next pairing.

### Conclusion and recommendation| 3.3

In conclusion the basic functionalities of the app are implemented. The app was designed using client server – model. The user interface itself was design base on the “simplify my life” design principle.  Some functionalities, such historical data were not implemented but are advanced enough in development to be implemented in future updates.

In order to ensure a stable app the application was tested.The components that were tested are speed, compatibilities (sdk 14 - 19), functionalities, response time and power usage.

The applications three most important function pairing algorithm, generating statistical information, player able to announce whether they are coming or not. These functionalities are all implemented and a work properly.

The group would recommend that the client improve the database on the server side. The client eventually needs to moving away from a one file solution to a normalized multitable database, this of course means that the client database will have to change as well.

# REFERENCES

1 Dr Peter Dauvergne (2002). The Case for Chess as a Tool to Develop Our Children’s Minds.

2 Peter Doggers (November 05, 2012). 75th Tata Steel tournament: players A & B group announced.

Retrieved from (2014) <http://www.chessvibes.com/announcement/75th-tata-steel-tournament-players-a-b-group-announced>

3 Harald Drillenburg (2014). Project Mobile Development

4 Wikipedia – last date modified:16 June 2014 at 22:45.- **Elo rating system**  
http://en.Wikipedia.org/wiki/Elo\_rating

5Senior Citizens and Digital Technology BY LEE RAINIE -Pewinternet.org – last modify setp 2012   
retrieve: juni 2014 <http://www.pewinternet.org/2012/09/15/senior-citizens-and-digital>

6 Tierstein, L.M. (1997). ["Managing a Designer/2000 Project"](http://www.wrsystems.com/whitepapers/managedes2k.pdf) (PDF). [New York Oracle User Group](http://www.nyoug.org/). Fall '97. Retrieved 2014-05-31.

7Rob Scheermeijer 2014- interview with the club leader.

Appendix A

Documentation of all Server files:

change\_absent\_admin.php:

requires: a secret key,hashed to identify the player that is doing the modificions, needs to match this players hashed secret key in the database

a state (true, false) to set the player to this state of attendance

(true=coming, false=not coming)

a player id, to identify the player to be changed

modifies: the database, updates in the players table all the attenance state

returns: true on succes, specific error messages on a missing parameter, a generic error on a serverside error

change\_absent\_personal.php:

requires a secret key, hashed to identify the player that is doing the modificions, needs to match this players hashed secret key in the database

a state (true, false) to set the modifying player to this state of attendance

(true=coming, false=not coming)

modifies: the database, updates in the players table all the attenance state

returns: true on succes, specific error messages on a missing parameter, a generic error on a serverside error

chessapp.php:

requires: a Kaizer file uploaded through the form on admin/index.php with the name KZ11 where 11 can be any combination of two integers.

modifies: checks all values in the Database, updates all tables that have different entrys between the Database and the Kaizer file.

Returns: a Successmessage on update, then automatically returns the user to the admin panel. Since the kaizer file is still beeing modified on the clients side no detailed error documentation exists.

export.php

requires: A Kaizer file must previously have been uploaded.

Modifies: The previously uploaded Kaizer File, changing the values for next weeks pairings, before outputting it as a download.

Returns: A download prompt for an updated Kaizer file.

upd\_player\_matches.php

requires: a unix timestamp (t), indicating the time of the last updates

modifies: -

Returns: all rows from all databases that have been modified since the last update in form of a json string in the format

{db\_name{[primary key], col1, col2, …,colN},

db\_name{[primary key],col1, col2, …, colN},

etc...}

verify.php

requires: a secret key,hashed to identify the player that is trying to sign into the app for the first time

modifies: -

returns: false, if this hashed secret key is not in the database

true, if the player is in the database

admin, if the player is in the database and flagged as admin

folder uploads

stores all uploaded Kaizer files for archiving purposes.

Folder Admin

contains the files changestuff.php and index.php

index.php

display:

displays all important information to the sites manager such as, including player id, first and last name, next weeks matchup/attendance, a display if a player permanently absent for this season, a display if a player is flagged as an admin and the secret key of each player used to unlock the app on the phone.

Players that are absent permanently, not admin or absent for the next meetup are displayed in red.

Players that are not permanently absent, are admin or are comingfor the next meetup are displayed in green.

Players that have a manual matchup or external game are displayed in yellow.

Functionality:

Clicking on the colored fields under the headers 'next pairings', 'permanent absent?' or 'admin?', changes those entries in the database.

(turning next\_pairing off if it is on or manually paired, or on if it is off.

And turning entries under 'permanent\_absent?' and 'admin?' on if they are off and vice versa.) In order to change an entry under 'next\_parings' to external or a manual matchup one has to click the link next to the column next\_pairings .

Changestuff.php

implements the ajax calls from index.php.

Requires: a type indicating wheter admin permission, permanent absent or next\_pairing are supposed to be changed.

A state indicating to wich state the database entry is supposed to be changed to.

a player id, to identify the player to be changed

modifies: The columns passed down in type, for the rows indicated with id into the state indicated with state.

Returns: true on succes, database error and false on a fault.

display:

displays all important information to the sites manager such as, including player id, first and last name, next weeks matchup/attendance, a display if a player permanently absent for this season, a display if a player is flagged as an admin and the secret key of each player used to unlock the app on the phone.

Players that are absent permanently, not admin or absent for the next meetup are displayed in red.

Players that are not permanently absent, are admin or are comingfor the next meetup are displayed in green.

Players that have a manual matchup or external game are displayed in yellow.

Functionality:

Clicking on the colored fields under the headers 'next pairings', 'permanent absent?' or 'admin?', changes those entries in the database.

(turning next\_pairing off if it is on or manually paired, or on if it is off.

And turning entries under 'permanent\_absent?' and 'admin?' on if they are off and vice versa.)

In order to change an entry under 'next\_parings' to external or a manual matchup one has to click the link next to the column next\_pairings and fill the matchup (or E for external) into the displayed promt box.

On top of the page there are two forms, one for uploading a new Kaizer file, and one for downloading the current, modified Kaizer file.

Changestuff.php

implements the ajax calls from index.php.

Requires: a type indicating wheter admin permission, permanent absent or next\_pairing are supposed to be changed or a manual/external pairing is made.

A state indicating to wich state the database entry is supposed to be changed to.

a player id, to identify the player to be changed

modifies: The columns passed down in type, for the rows indicated with id into the state indicated with state.

Returns: true on succes, database error and false on a fault.

Documentation of all Server files:

change\_absent\_admin.php:

requires: a secret key,hashed to identify the player that is doing the modificions, needs to match this players hashed secret key in the database

a state (true, false) to set the player to this state of attendance

(true=coming, false=not coming)

a player id, to identify the player to be changed

modifies: the database, updates in the players table all the attenance state

returns: true on succes, specific error messages on a missing parameter, a generic error on a serverside error

change\_absent\_personal.php:

requires a secret key, hashed to identify the player that is doing the modificions, needs to match this players hashed secret key in the database

a state (true, false) to set the modifying player to this state of attendance

(true=coming, false=not coming)

modifies: the database, updates in the players table all the attenance state

returns: true on succes, specific error messages on a missing parameter, a generic error on a serverside error

chessapp.php:

requires: a Kaizer file uploaded through the form on admin/index.php with the name KZ11 where 11 can be any combination of two integers.

modifies: checks all values in the Database, updates all tables that have different entrys between the Database and the Kaizer file.

Returns: a Successmessage on update, then automatically returns the user to the admin panel. Since the kaizer file is still beeing modified on the clients side no detailed error documentation exists.

export.php

requires: A Kaizer file must previously have been uploaded.

Modifies: The previously uploaded Kaizer File, changing the values for next weeks pairings, before outputting it as a download.

Returns: A download prompt for an updated Kaizer file.

upd\_player\_matches.php

requires: a unix timestamp (t), indicating the time of the last updates

modifies: -

Returns: all rows from all databases that have been modified since the last update in form of a json string in the format

{db\_name{[primary key], col1, col2, …,colN},

db\_name{[primary key],col1, col2, …, colN},

etc...}

verify.php

requires: a secret key,hashed to identify the player that is trying to sign into the app for the first time

modifies: -

returns: false, if this hashed secret key is not in the database

true, if the player is in the database

admin, if the player is in the database and flagged as admin

folder uploads

stores all uploaded Kaizer files for archiving purposes.

Folder Admin

contains the files changestuff.php and index.php

index.php

display:

displays all important information to the sites manager such as, including player id, first and last name, next weeks matchup/attendance, a display if a player permanently absent for this season, a display if a player is flagged as an admin and the secret key of each player used to unlock the app on the phone.

Players that are absent permanently, not admin or absent for the next meetup are displayed in red.

Players that are not permanently absent, are admin or are comingfor the next meetup are displayed in green.

Players that have a manual matchup or external game are displayed in yellow.

Functionality:

Clicking on the colored fields under the headers 'next pairings', 'permanent absent?' or 'admin?', changes those entries in the database.

(turning next\_pairing off if it is on or manually paired, or on if it is off.

And turning entries under 'permanent\_absent?' and 'admin?' on if they are off and vice versa.) In order to change an entry under 'next\_parings' to external or a manual matchup one has to click the link next to the column next\_pairings .

Changestuff.php

implements the ajax calls from index.php.

Requires: a type indicating wheter admin permission, permanent absent or next\_pairing are supposed to be changed.

A state indicating to wich state the database entry is supposed to be changed to.

a player id, to identify the player to be changed

modifies: The columns passed down in type, for the rows indicated with id into the state indicated with state.

Returns: true on succes, database error and false on a fault.

display:

displays all important information to the sites manager such as, including player id, first and last name, next weeks matchup/attendance, a display if a player permanently absent for this season, a display if a player is flagged as an admin and the secret key of each player used to unlock the app on the phone.

Players that are absent permanently, not admin or absent for the next meetup are displayed in red.

Players that are not permanently absent, are admin or are comingfor the next meetup are displayed in green.

Players that have a manual matchup or external game are displayed in yellow.

Functionality:

Clicking on the colored fields under the headers 'next pairings', 'permanent absent?' or 'admin?', changes those entries in the database.

(turning next\_pairing off if it is on or manually paired, or on if it is off.

And turning entries under 'permanent\_absent?' and 'admin?' on if they are off and vice versa.)

In order to change an entry under 'next\_parings' to external or a manual matchup one has to click the link next to the column next\_pairings and fill the matchup (or E for external) into the displayed promt box.

On top of the page there are two forms, one for uploading a new Kaizer file, and one for downloading the current, modified Kaizer file.

Changestuff.php

implements the ajax calls from index.php.

Requires: a type indicating wheter admin permission, permanent absent or next\_pairing are supposed to be changed or a manual/external pairing is made.

A state indicating to wich state the database entry is supposed to be changed to.

a player id, to identify the player to be changed

modifies: The columns passed down in type, for the rows indicated with id into the state indicated with state.

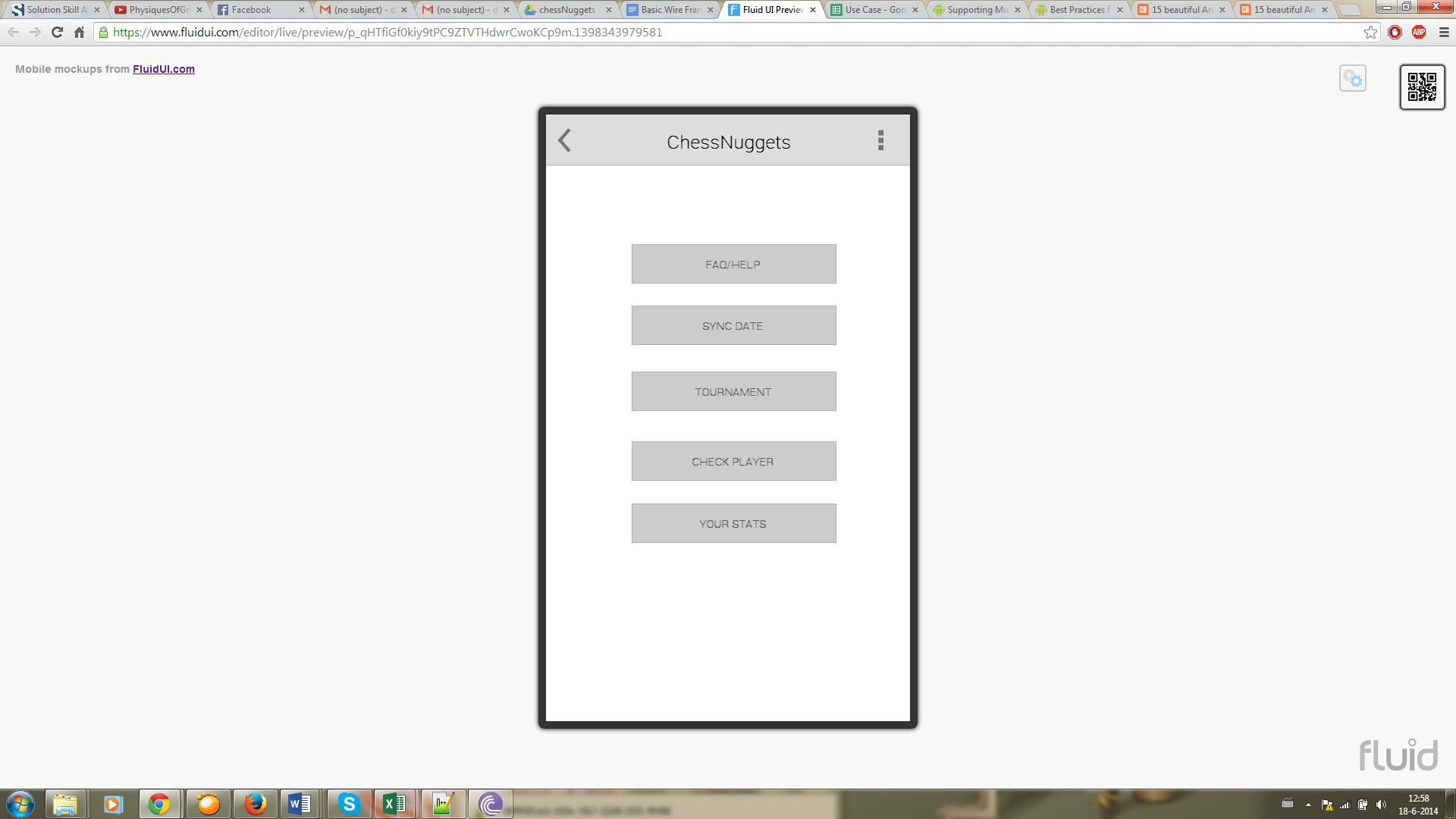
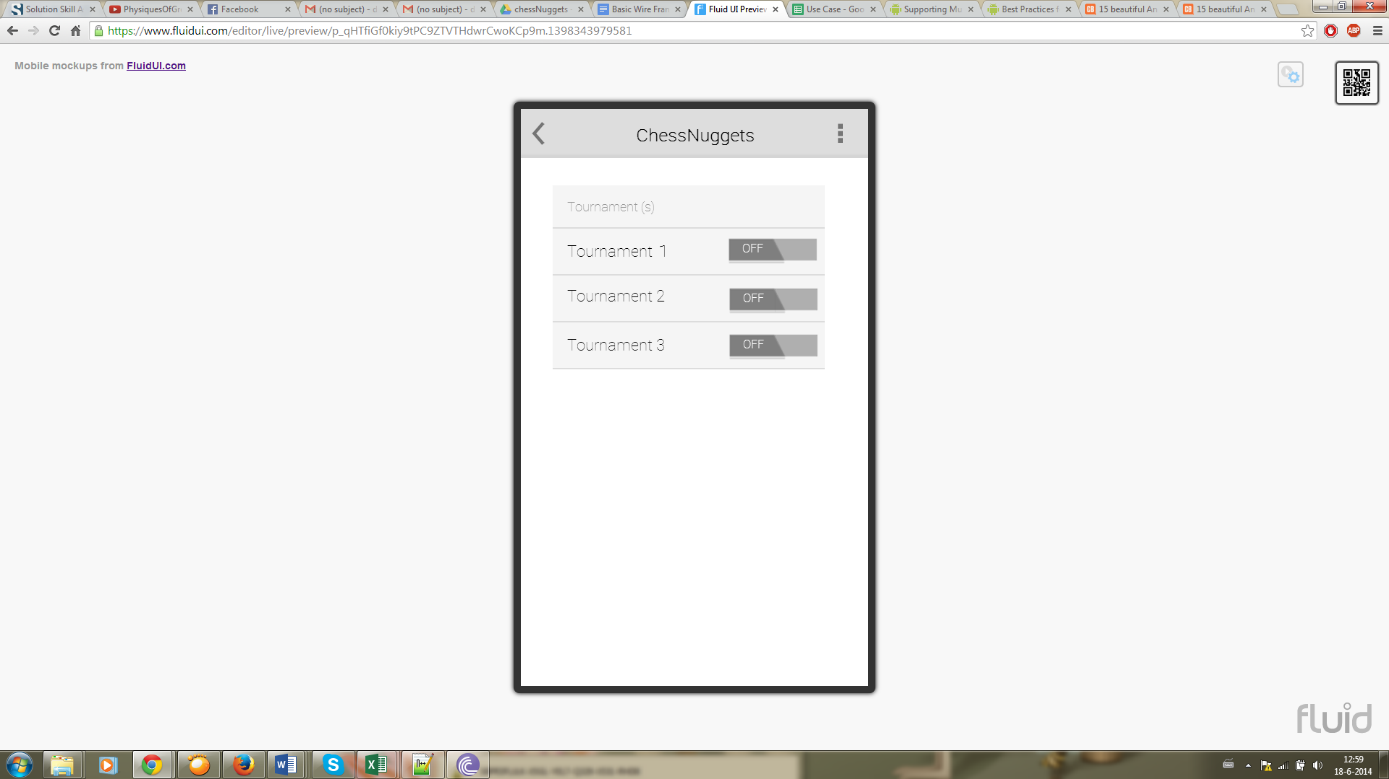
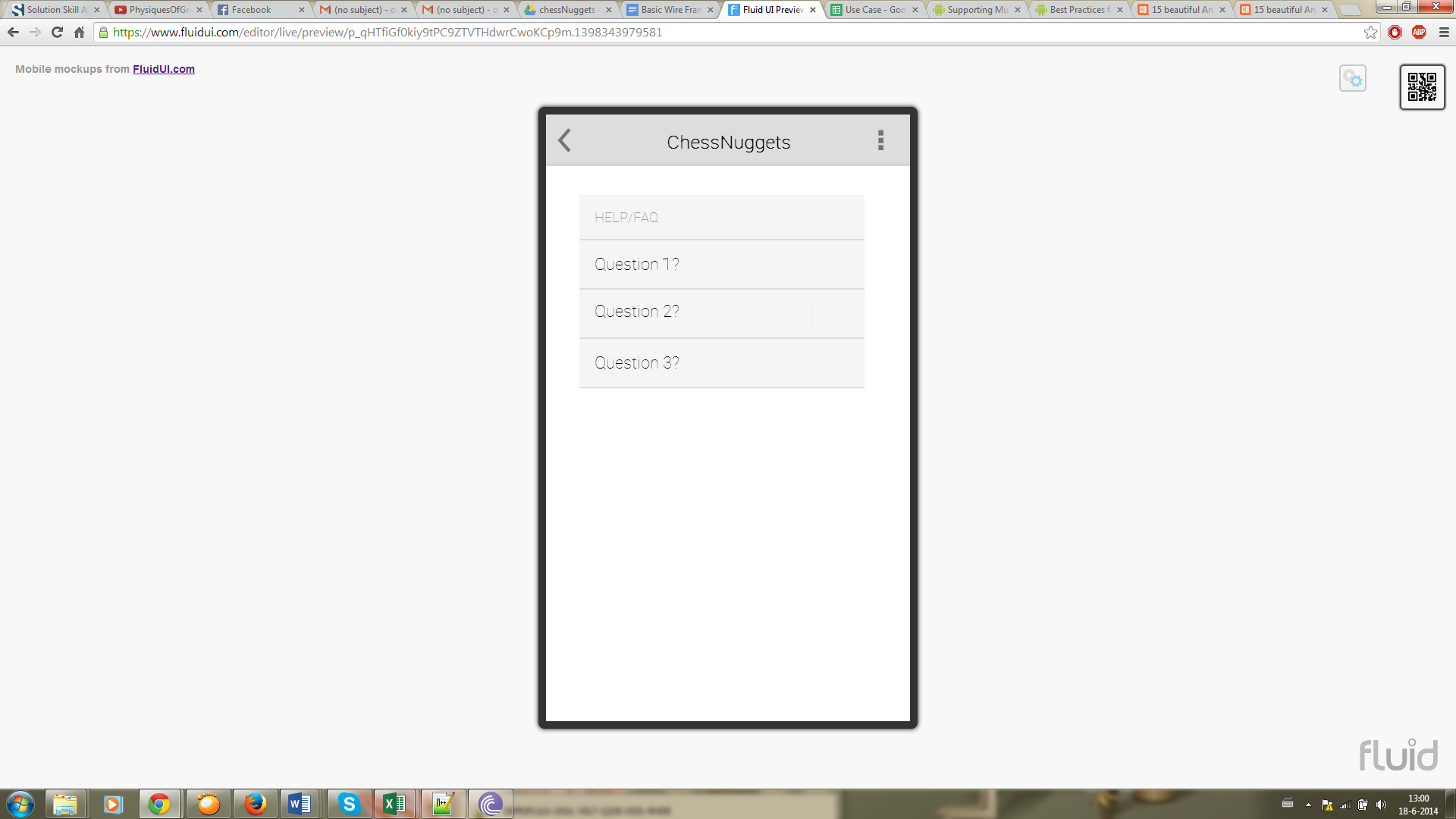
Returns: true on succes, database error and false on a fault.

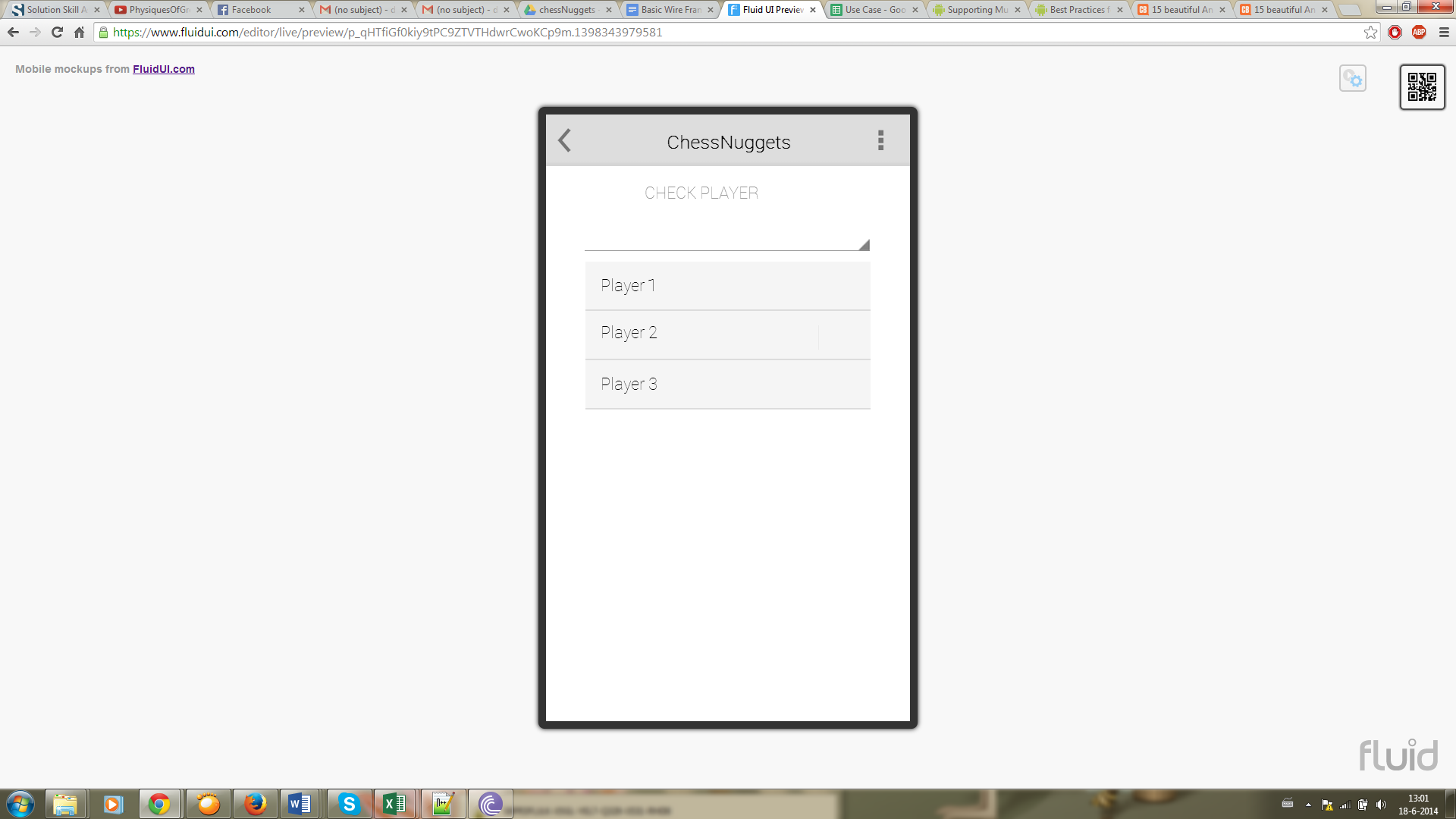
Appendix B

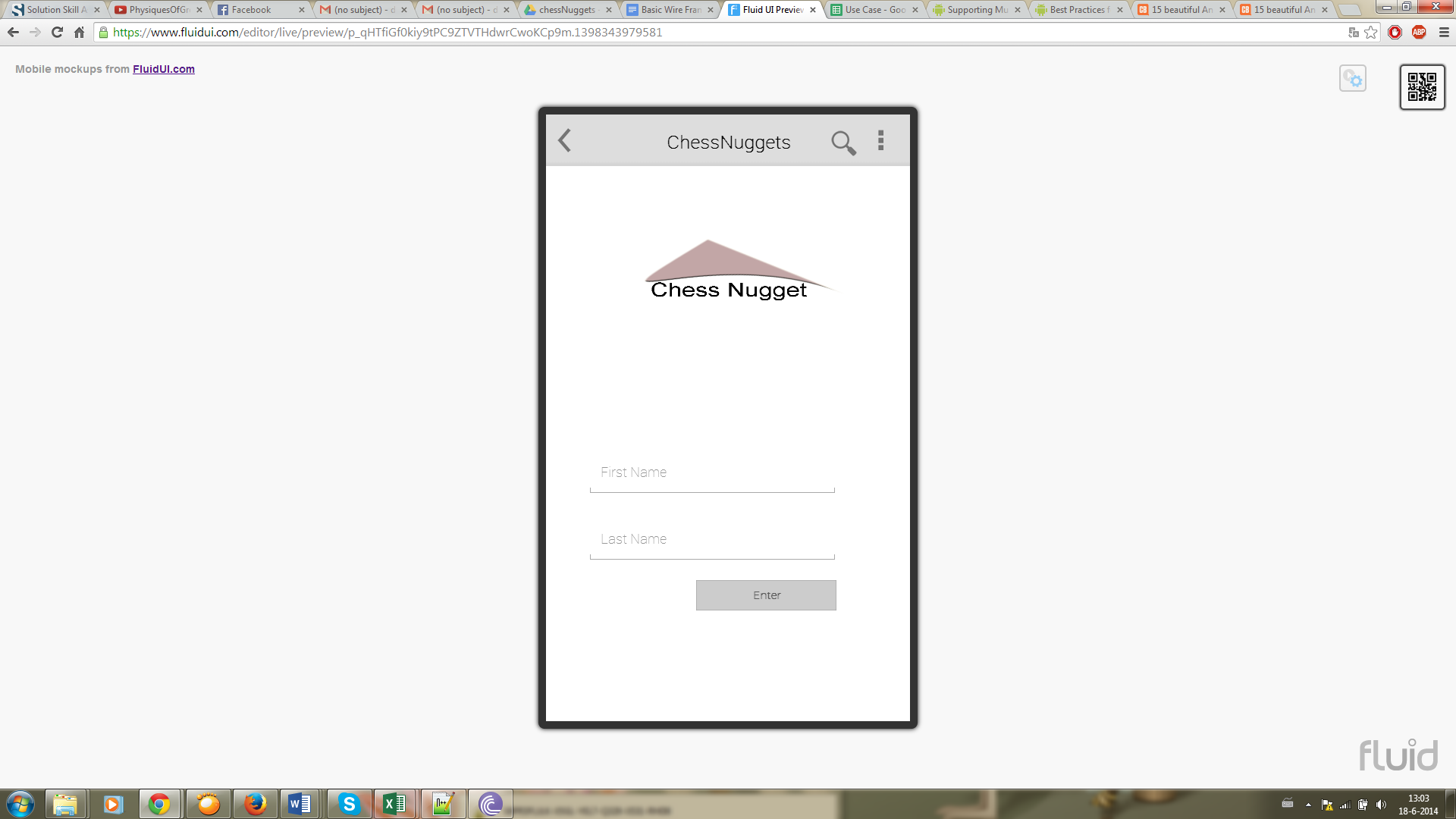
|  |  |  |
| --- | --- | --- |
| **B** | Sink data | |
| **Description** | Player(chess Tournament Player) provides link to sink data | |
| **Used by** | Used by Use Case 2 | |
| **PreConditions** | Players details are uptodate | |
| **Success End Conditions** | System will store players details | |
| **Failed End Conditions** | No data | |
| **Actors** | Player, system(chess Nugget) | |
| **Trigger** | Player Details(Use Case 2) | |
| **Description** | **Step** | **Action** |
|  | 1 | Player clicks on Sink Data Button |
|  | 2 | Player inputs the Url |
|  | 3 | System looks up for required data |
|  | 4 | Stores Data |
|  |  |  |
| **Extensions** | **Step** | **Branching Action** |
|  | 3 | System ask to re-enter URL if not found |
|  |  |  |
| **Variations** |  | **Branching Action** |
|  |  | N/A |
|  |  |  |
| **Exceptions** |  | No data found , End of Use case 1 |
|  |  |  |
| **Additional Information** |  | |
| **OPEN ISSUES** |  | |
| **Due Date** | release 1.0 | |

|  |  |  |
| --- | --- | --- |
| **USE CASE 2** | Chess tournament player(Player) details | |
| **Description** | Player inputs his name to view details | |
| **Used by** | N/A | |
| **PreConditions** | Player have their rankings and details on the data that was sinked | |
| **Success End Conditions** | Player will be able to view his details according to the data sinked | |
| **Failed End Conditions** | Player wont see the details, Chess tournament player Inputs details through system as stand alone | |
| **Actors** | Player, system(chess Nugget) | |
| **Trigger** | Tournaments Details | |
| **Description** | **Step** | **Action** |
|  | 1 | Players inputs name |
|  | 2 | System looks for pre sinked data to find the user name and populate the details |
|  |  |  |
| **Extensions** | **Step** | **Branching Action** |
|  | 2 | player details cannot be found, system provides with an option to add data |
|  |  | |
| **Variations** |  | **Branching Action** |
|  |  | N/A |
|  |  |  |
| **Exceptions** |  | Player detail not entered end of use case 2 |
|  |  |  |
| **Additional Information** | Data need to be sinked that contains all details of Players | |
| **OPEN ISSUES** |  | |
| **Due Date** | release 1.0 | |

Appendix C

# Appendix D Meeting Minutes

April 17th, 2014

## Attendees

Nirajan Pokharel(523026),Tobias Winkels-Herding (521640),Kingson,Daniel

## Scribe

Nirajan Pokharel(523026)

## Agenda/minutes

* Discussion about how to get a client:Since we didn’t had client yet for our project, Our teacher Dillenburg provided us an idea how we can success .
* Discussion about Documentation:We must produce 3 different types of document

User Manual (functional )

Developer Document(Java doc can be used)

Process Documentation(A detailed REport)

* Discussion regarding how to fix meeting with mentor: Send invitation with agenda and minutes next week 24 hours before the meeting actually starts.

## Next Meeting

April 24th, 2014

## Next Meeting Agenda Items

## Adjournment

NON

# Meeting Minutes

April 24th, 2014

## Attendees

Nirajan Pokharel(523026),Tobias Winkels-Herding (521640),Kingson,Daniel

## Scribe

Nirajan Pokharel(523026)

## Agenda for this meeting

* Discussion about Client
* steps that we can take before we get updates from our client about required functionality.

## Next Meeting

April 24th, 2014

## Meeting minutes

## **Help regarding minutes will be provide by Belinda in English 5 class,and Tobas will attend the class**

* Client is happy to use the app.
* Discussion regarding how to successfully complete the assignment(diplomacy /politics).
* why we need GP(General public license)
* Git set up
* client meeting to catch up with requirement.
* preparation of use case by Nirajan and Wire frame by Kingson.
* Requirement - Swiss System .

## Adjournment

NON

**Agenda for next meeting(**8th may 2014**)**

* Feedback from Client.
* Git set up.
* show use case and Wireframes.

# Meeting Minutes(WEEK 2nd)

held on **April 8th, 2014**

## Attendees

Nirajan Pokharel(523026),Tobias Winkels-Herding (521640),Kingson,Daniel

## Scribe

Nirajan Pokharel(523026)

## Agenda for this meeting

* Feedback from Client.
* Git set up.
* show use case and Wireframes.

## Meeting minutes

## meeting with the client to be held on thursday, April 8th, 2014, and the feedback from them to be discussed in upcoming meeting 15th may 2014.

* Git setup performed by Tobias, providing access was the other task, to be performed.
* Prepared use cases and presented during meeting.

## Next Meeting

April 15th, 2014

## Adjournment

NON

**Agenda for next meeting(**15th may 2014**)**

* Discussion about what client said last thursday, **April 8th, 2014**
* Discussion about the Gui prepared for the App,
* Discussion about ideas developed to sink ratings from the client .

# Meeting Minutes(WEEK 5th)

held on April 24th, 2014

## Attendees

Nirajan Pokharel(523026),Tobias Winkels-Herding (521640),Kingson,Daniel

## Scribe

Nirajan Pokharel(523026)

## Agenda for this meeting

* show prepared implementation to teacher.
* discussion about the implemented functionality and ways we are working with data (json string)
* get feedback from teacher.
* Report structure regarding the research questions

## Meeting minutes

* discussion about the json string , databases (structure should be normalised)
* json string to be used as data source in structured way
* use a global variables.

## Next Meeting

April 8th June, 2014(next week)

## Adjournment

NON

**Agenda for next meeting(**8th june 2014**)(WEEK 6th)**

* Discussion About all the progress made so far
* Discussion about what more steps are involved to successfully complete the assignment/project.

# Meeting Minutes(WEEK 6th)

held on June 5th, 2014

## Attendees

Nirajan Pokharel(523026),Tobias Winkels-Herding (521640),Kingson,Daniel

## Scribe

Nirajan Pokharel(523026)

## Agenda for this meeting

* Discussion About all the progress made so far
* Discussion about what more steps are involved to successfully complete the assignment/project.

## Meeting minutes

* sign in and sign out functionality (discussion)
* showed design to teacher
* showed app to teacher
* invite clients for meting

## Next Meeting

only final presentation

## Adjournment

NON

**User manual Server Side**

The Applications Server side is easy to manage, however since it interfaces directly with the Database potential mistakes can have huge consequences.

The most important thing to note is to not overwrite the Kaizer file multiple times in the same week.

The upload/download mechanism on top of the page are designed to receive the kaizer File on thursday after the chessclubs meetup, and to return a modified Kaizer file the following thursday before the next matchmaking takes place.

**Should the Kaizer file be uploaded multiple times during the week it will modify the database in a way that it will be returned to the state of the latest uploaded file. All changes made between the upload and the reupload of the file are forgotten.**

Other functionalities:

All colored fields are clickeable.



Clickingon them changes their state either to green or red.

Green for the column perm\_absent means that the player is not permanently absent,

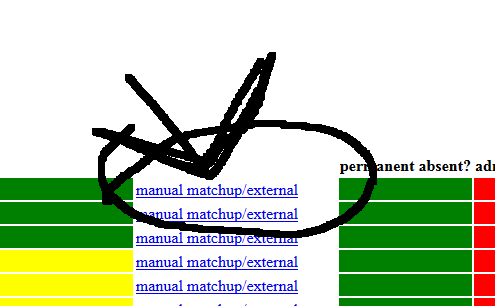
Green for the column admin means that the player iscurrently an admin.

Red (A) in the column next\_pairing means the player is not participipating in next weeks meetup.

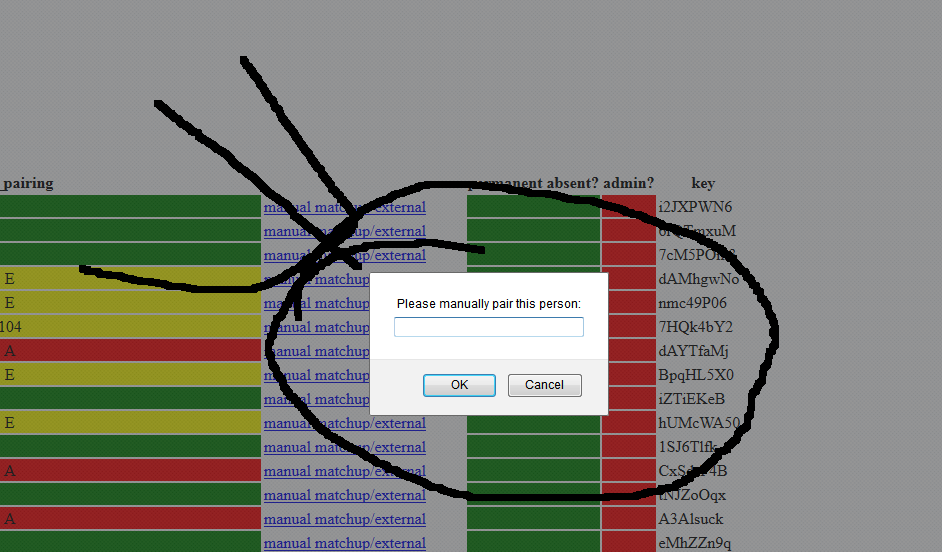
Green() in the column next\_pairing means the player is participipating in next weeks meetup.

Yellow([four letters]) in the column next\_pairing means the player is manually matched, either internal or external or has other club based activites.

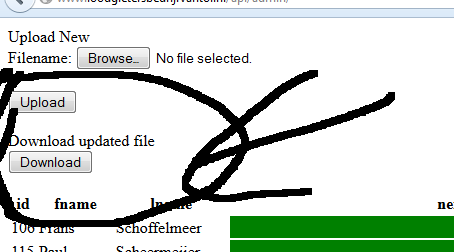
**In order to change a players status to manually matched the link with the text manual matchup/external can be clicked in the corresponding players row.**



**A message promt will appear that requires the administrator to enter a Kaizer conform four letter combination. This combination will not be doublechecked by the server but instead will be inserted in the Database blindly.**



**It is task of the Administrator to make sure that the combination is indeed correct.**

Downloading a new Kaizer file. This will return a modified version of last weeks file, the modifications are limited to the pairingsas these will be used to calculate the next opponents.