HR Páver System Design

Verklegt Námskeið 2

Birkir F. Baldursson
Egill A. Hlöðversson
Kristófer A. Kristinsson
Ragnar M. Heinesen
Ríkarður Einarsson

Spring 2015

Table of Contents

Hópur 37

May 5, 2015

Contents

1 Introduction

1.1 Overview

The goal of **HR Páver** is to improve interactions between students and staff within Reykjavík University. Our system will strive to be more convenient in use than the current systems. The purpose of this system design is to map out the user interface, basic flow of the system and define our database structure.

1.2 Definitions and acronyms

HR Páver is the name of our system.

RU stands for **R**eykjavík **U**niversity

2 Technical information

2.1 Operations environment

The system is designed to combine functions of current systems used by students and staff of Reykjavík University into a single website. The site will be used by aforementioned groups and if a user wishes to register he must have a RU email address. The system will run on a Windows Server 2012 which includes Internet Information Server 8 (IIS 8) and SQL Server 2014. Anyone who has a RU email address, an internet connection and a web browser will be able to use the website. The website shall use responsive bootstrap design so that it will be available on screens with varying resolutions.

2.2 Development environment

The system design report is written using www.shareLaTeX.com Our website is built using the development environment Microsoft Visual Studio 2013. The web application framework used is ASP.NET with the programming language C#. www.Github.com is used for collaboration and source code management. Additionally we use Google drive to store miscellaneous files including a work schedule log. Balsamiq is used to make low-fi prototypes. SQL management in visual studio is used to manage the database. Five laptops are used in the development of this assignment. Four of the laptops use the operating system windows 8 and one uses Mac OS X and all of them have Visual Studio 2013 installed. The browsers used for testing are Chrome, Safari, Firefox and IE.

3 Rules

3.1 General rules

User cannot register a new account without a @ru.is email address.

User must be logged in to use the system.

User cannot upload pictures larger than 2MB.

User cannot upload a profile picture smaller than 50x50px.

User cannot use html code in posts, usernames or description.

User cannot use SQL code in posts, usernames or description.

User cannot create an event with a start date that has already passed.

3.2 Programming rules

Well designed system usually include ordered rules for the code. This helps programmers coordinate when writing code for a new system. Rules are important for a programmer because they can:

Create consistent look to the code.

Enable readers to understand the code more quickly.

Facilitate changing and maintaining of the code.

Here we will list the coding rules we intend to follow.

3.2.1 Layout rules

- Use Source code style.
- Smart indenting.
- One statement per line.
- One declaration per line.
- One blank line between method definitions and property definitions.
- Use curly braces around blocks, even when they would be optional.

3.2.2 Naming rules

- Using statements are placed at the top of the file, before the namespace declaration, with system directive first.

3.2.3 Commenting rules

- Comments begin with an uppercase letter.
- One space between the comment delimiters (//) and the comment text.
- Comments are placed above the code not by its side.

3.2.4 Variable rules

- Using var is okay, if and only if the type is obvious.

3.2.5 Error handling rules

- Use a try-catch statement for most exception handling.

3.2.6 Operator rules

- Use (&&) and (||) operators when comparing to avoid exceptions and increase performance.

3.2.7 LINQ rules

- Use meaningful names for query variables.
- Use Pascal casing for property names.
- Use implicit typing in the declaration of query variables and range variables.

3.2.8 Javascript rules

- Namespaces and Classes are Pascal cased.
- Properties, fields, local variables and parameters are Camel cased.
- Functions are Camel cased if and only if they are not class constructors then they are Pascal cased.

3.3 Interface rules

While designing the interface of our website, our guidelines will be the two key principles by Donald Norman,[1] visibility and affordance. The eight golden rules by Ben Schneiderman[2] will also be kept in mind. Colours associated with Reykjavík University are red and orange-red. We feel red is too aggressive for a social website so we decided to use orange, because of its connection with communication and it can be found in the RU logo. For readability, other colours in our palette will be light grey, white and black. Our website will use two types of sound alerts, one jolly but mellow for notifications and the other alerting but not too angry. The interface will have a responsive design for it to be accessible on mobile and desktop platforms.

Text Rules

- Bootstrap will control heading size.
- Headings shall not end with a period.
- Bootstrap will control body text size.
- Font family will be used in this order Helvetica, Arial, Calibri.
- Dates will be represented in European style dd/mm/yyyy.

[1]http://architectingusability.com/2012/06/28/donald-normans-design-principles-for-usability/
[2]http://faculty.washington.edu/jtenenbg/courses/360/f04/sessions/schneidermanGoldenRules.html

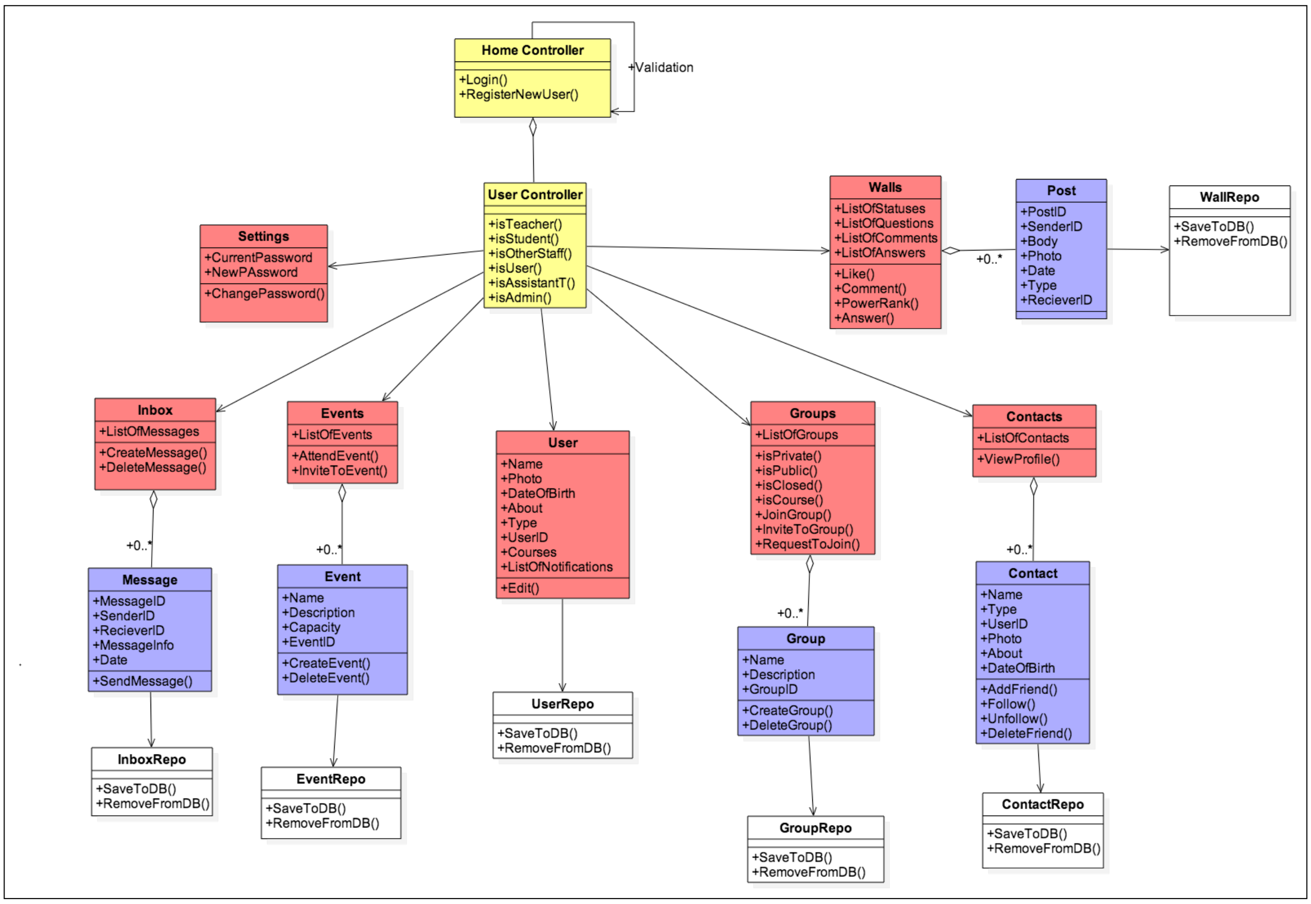
4 Component design

4.1 Class diagram

Class diagram describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships between objects.

The colours in this diagram split the classes into layer groups.

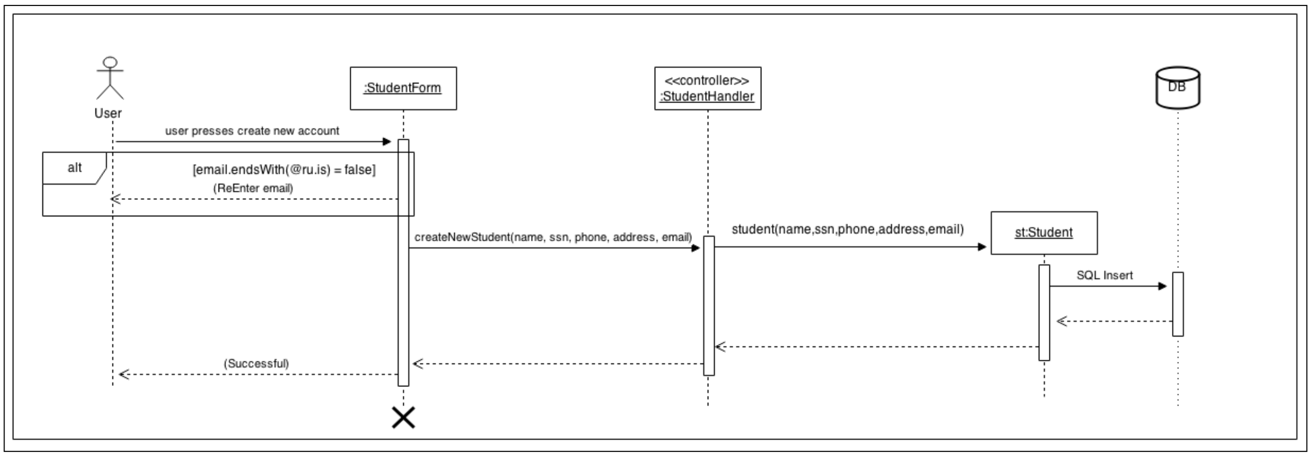
- Yellow = Controllers
- Red = Models
- Blue = Views
- White = Repositories



Class diagram

4.2 Sequence diagram

A sequence diagram is a form of interaction diagram which shows objects as lifelines running down the page, with their interactions over time represented as messages drawn as arrows from the source lifeline to the target lifeline.[3]


User
:StudentForm
<<controller>>
:StudentHandler
DB
user presses create new account
alt
[email.endsWith(@ru.is) = false]
(ReEnter email)
createNewStudent(name, ssn, phone, address, email)
student(name,ssn,phone,address,email)
st:Student
SQL Insert
(Successful)

Sequence diagram of registering a new account

4.3 State chart diagrams

State diagrams are used to give an abstract description of the behavior of a system.[4] They are used to describe systems that have dynamic behaviours and often used to describe the state of a particular class.

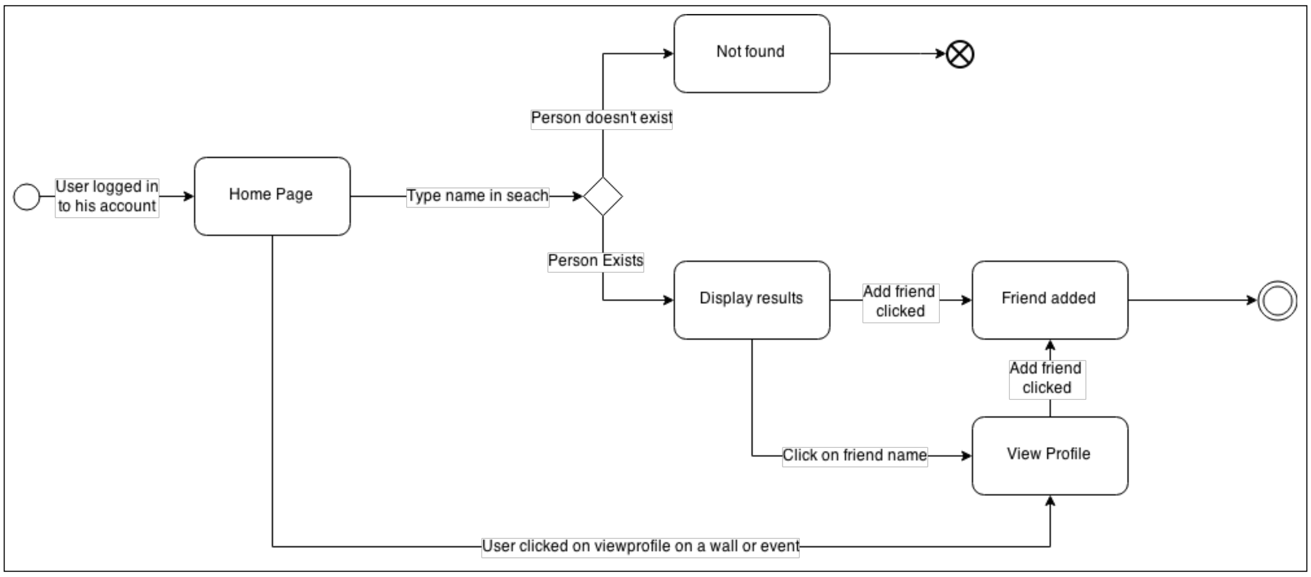


Diagram of adding a friend.

[3]http://www.sparxsystems.com/resources/uml2_tutorial/uml2_sequencediagram.html
[4]http://en.wikipedia.org/wiki/State_diagram

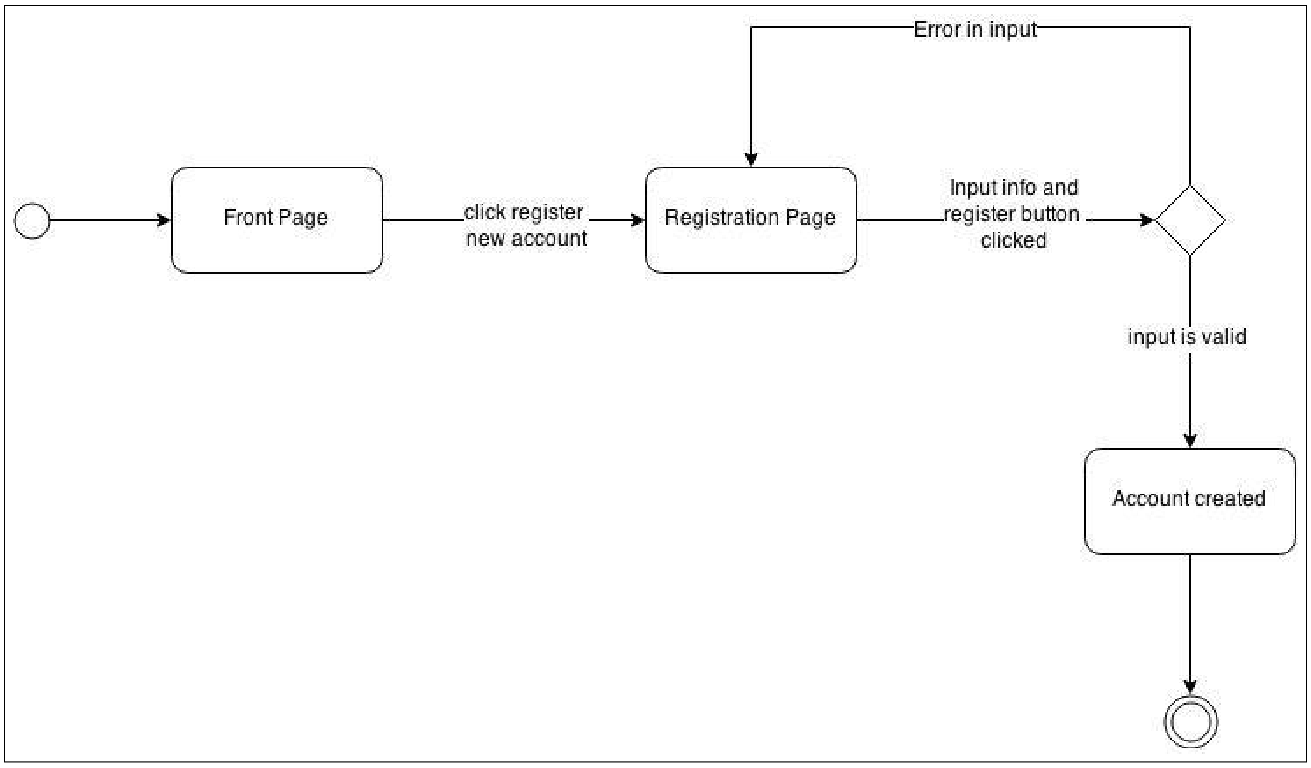


Diagram of registering a new account.

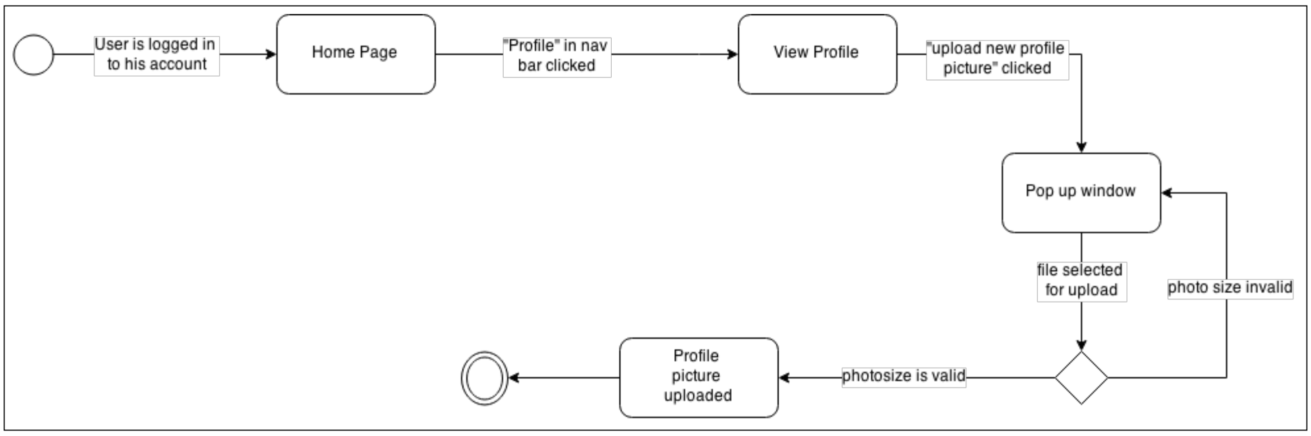


Diagram of uploading a profile picture.

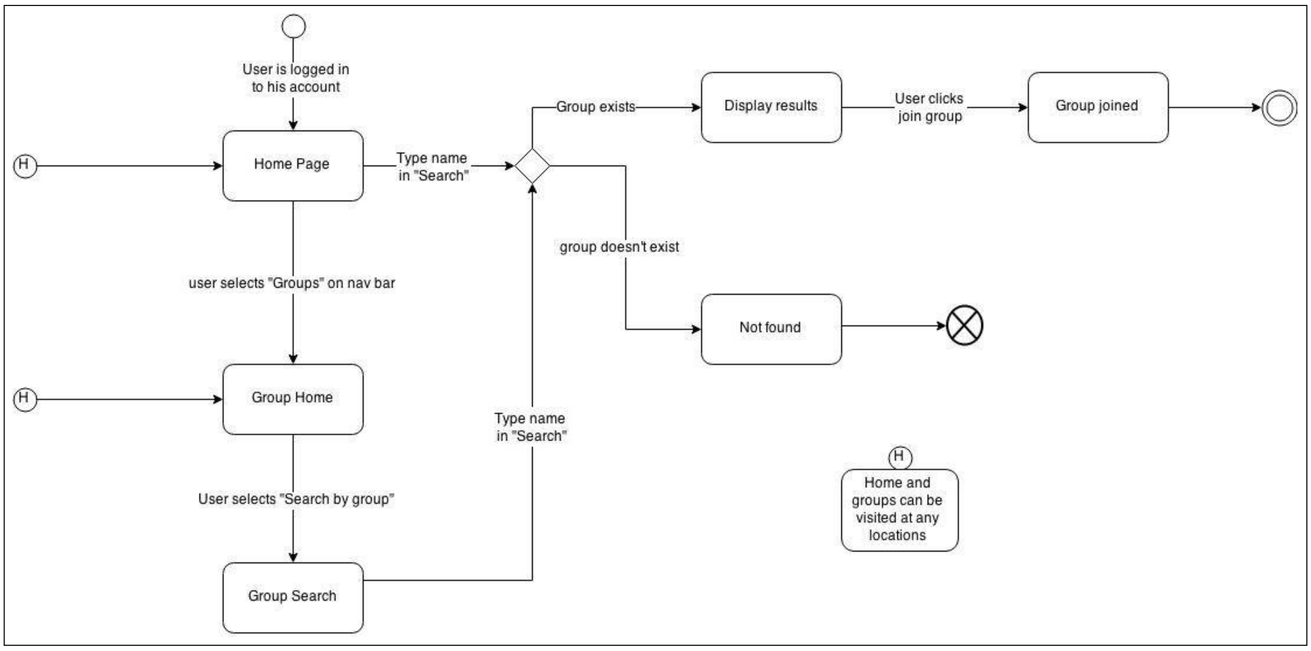
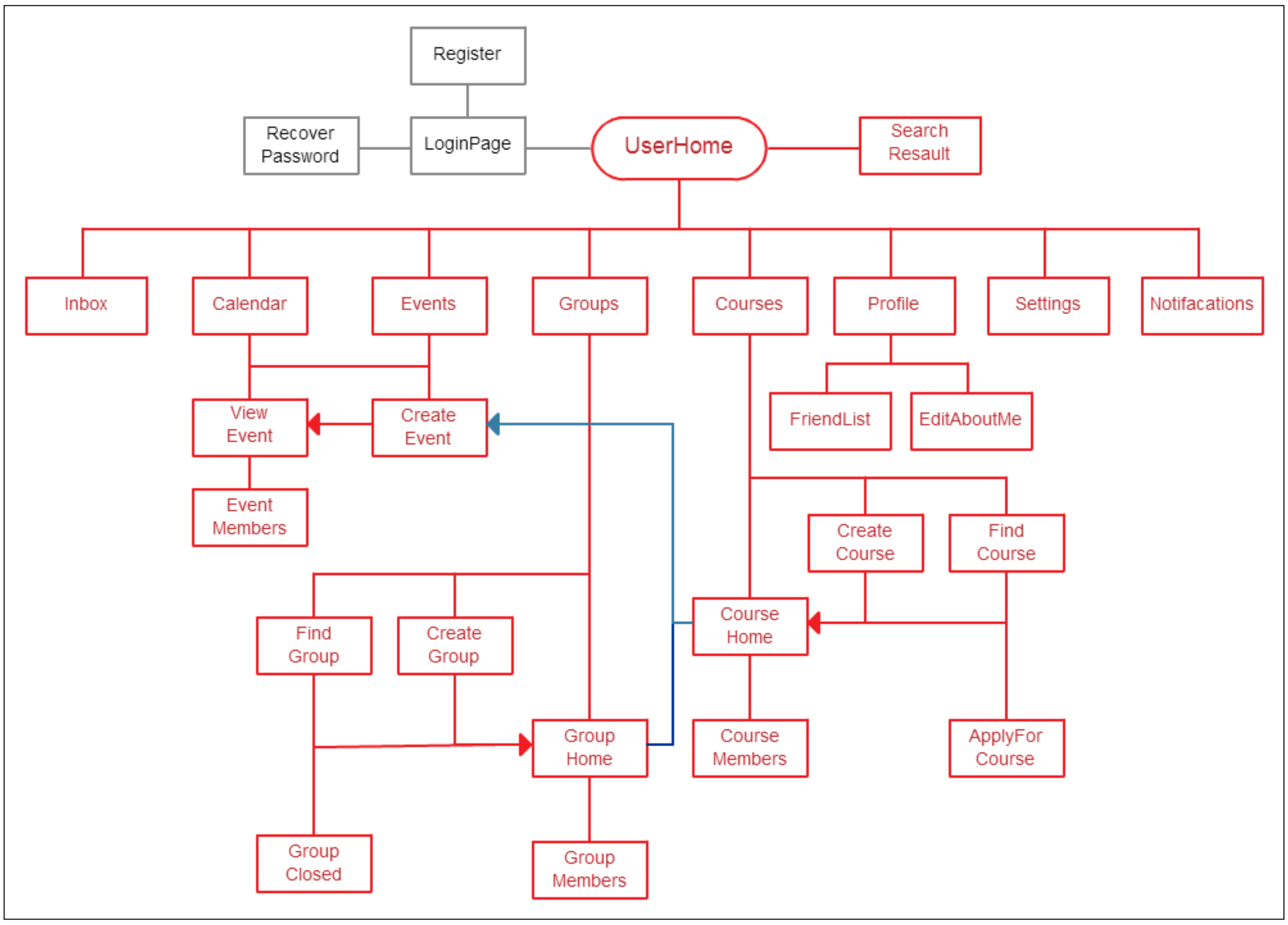


Diagram of searching for a public group.

5 User interface design

5.1 Navigation diagram

The purpose of the navigation diagram below is to illustrate a low level architecture. It demonstrates how a user can navigate around the website.

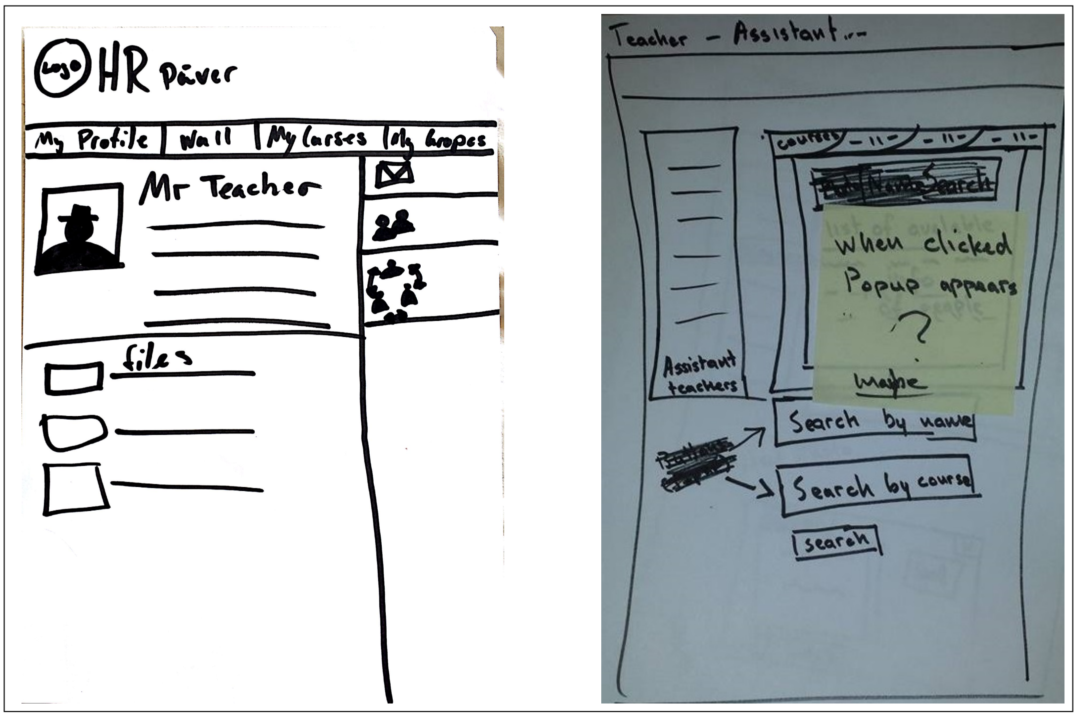


Navigation diagram.

5.2 Low fidelity prototypes

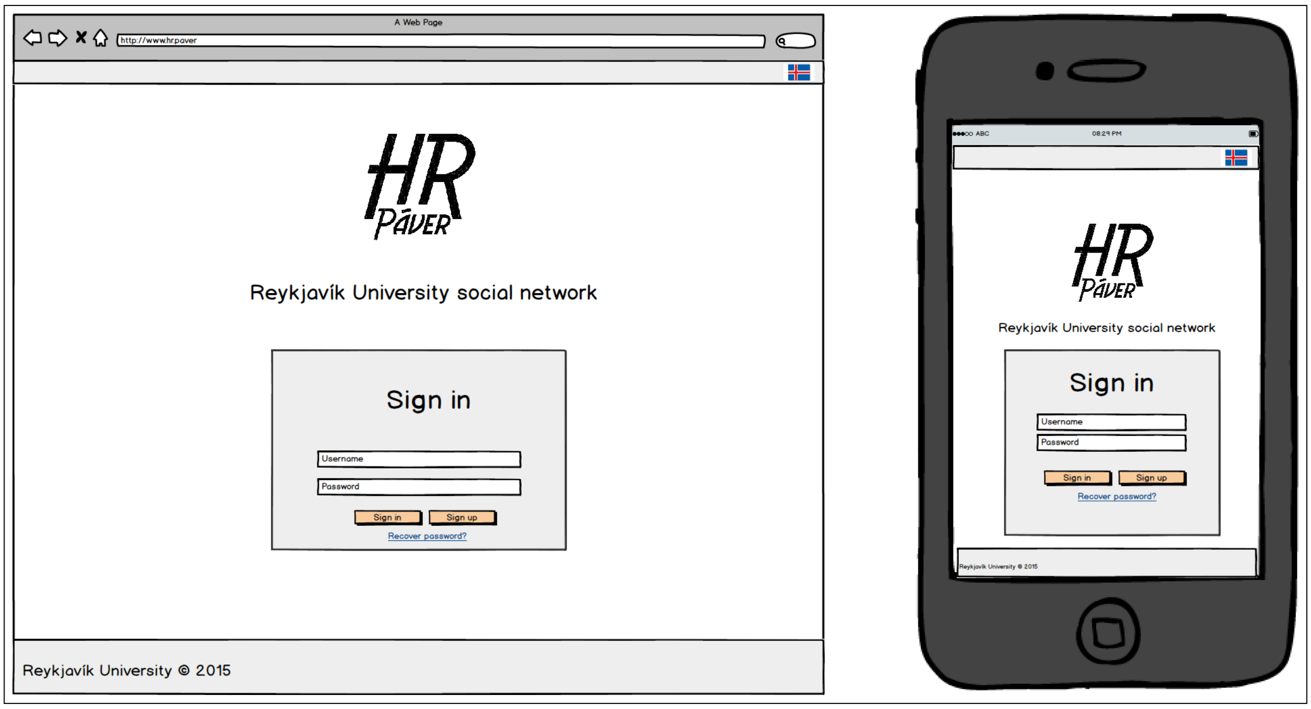
Low-fidelity prototype is built because it can be quickly exposed to user feedback. That enables us to visualize and solve core issues related to the product's usability and proposed functionality. We start off by sketching various ideas. Here are a couple of our sketch ideas:

5.2.1 Sketches

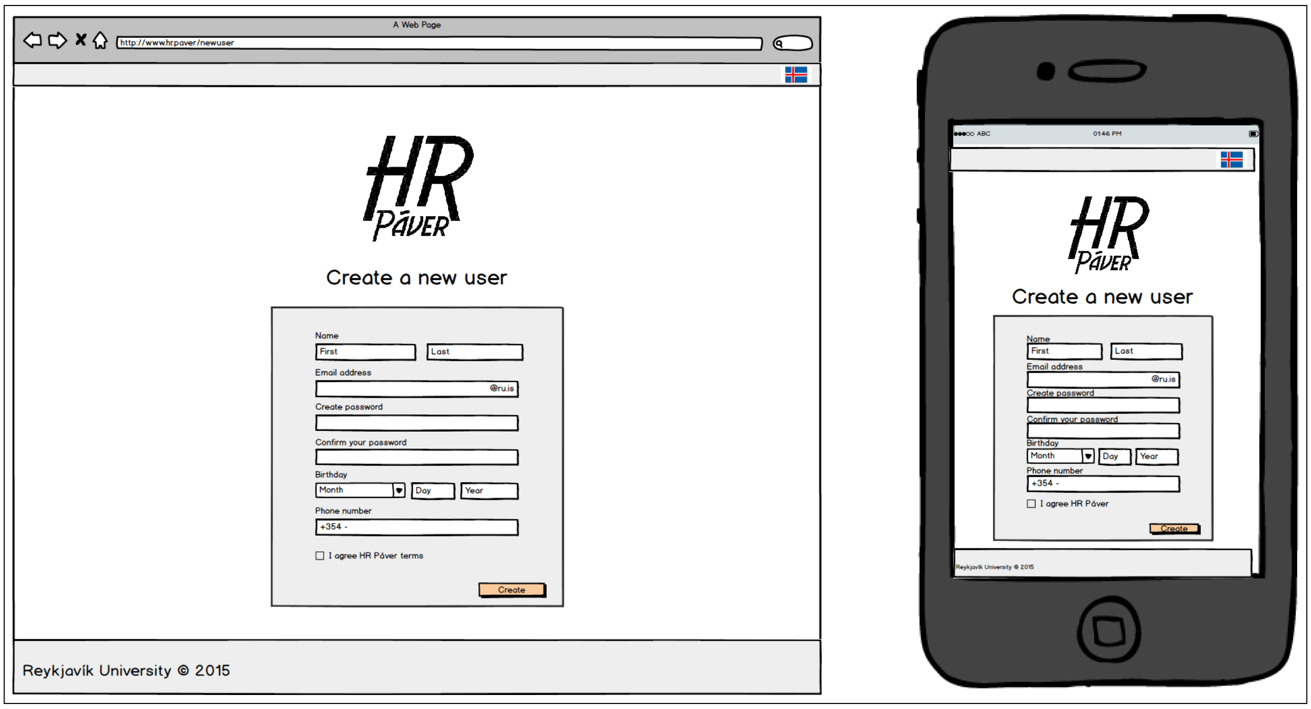


LO-FI My Profile & Assistant Teacher Groups

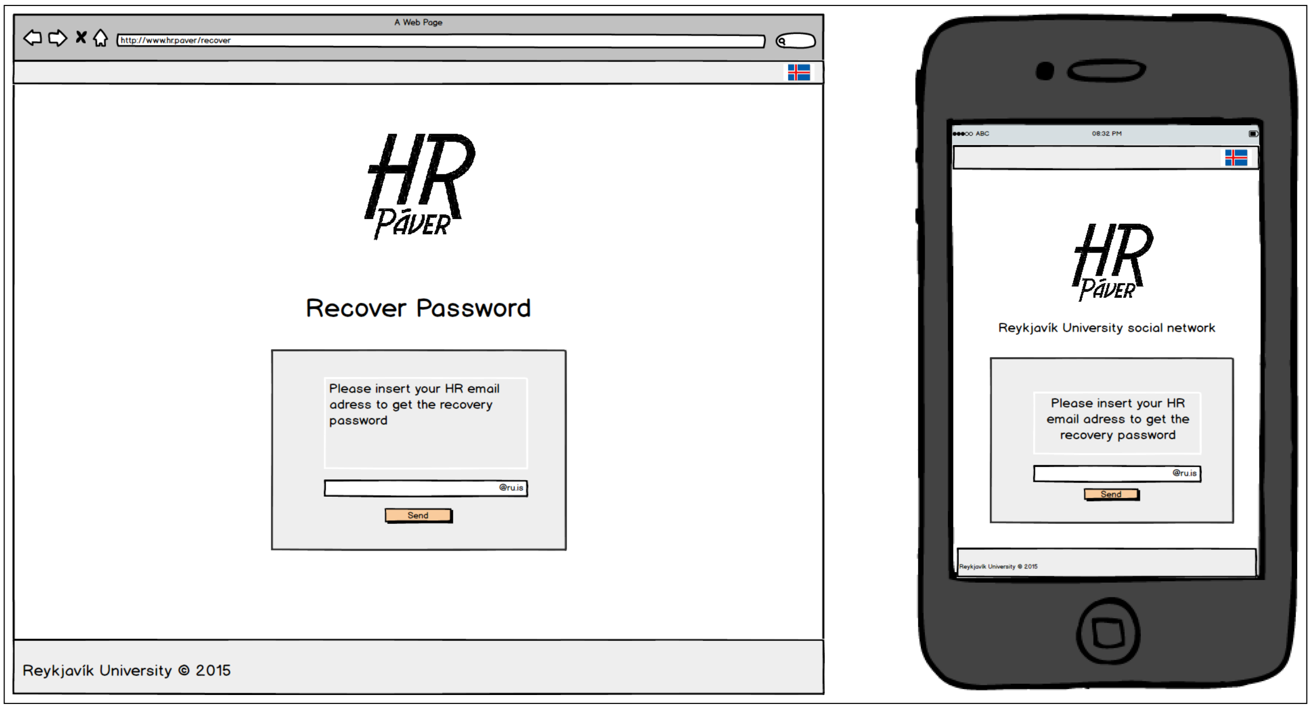
5.2.2 Final prototype



Front Page
The idea was to have the homepage as welcoming as possible. The only condition is that the user has to input his RU email and password.

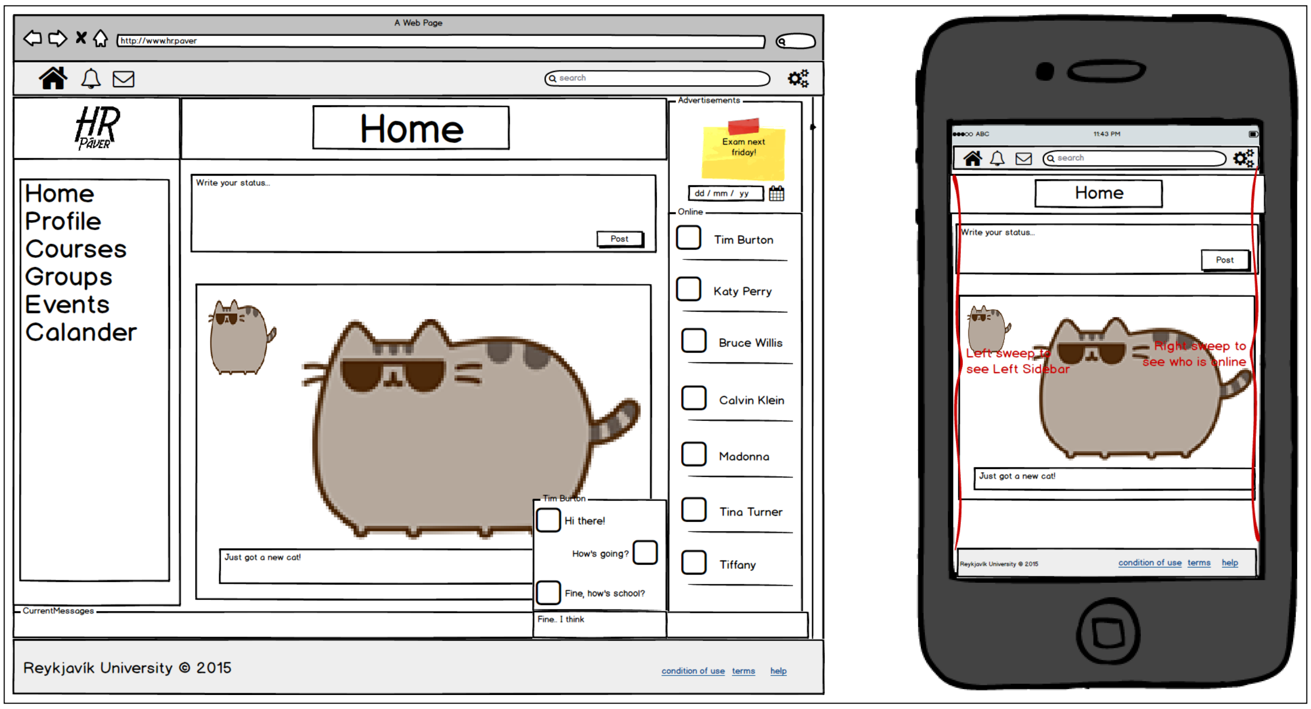


Registering new users
When registering, all fields are required to be filled out with valid input.The user must also agree to HR Páver terms and conditions.


A Web Page
http://www.hrpaver/recover
HR PÁVER
Recover Password
Please insert your HR email adress to get the recovery password
@ru.is
Send
Reykjavík University © 2015
ABC
08:32 PM
Reykjavík University social network
Reykjavík University © 2015

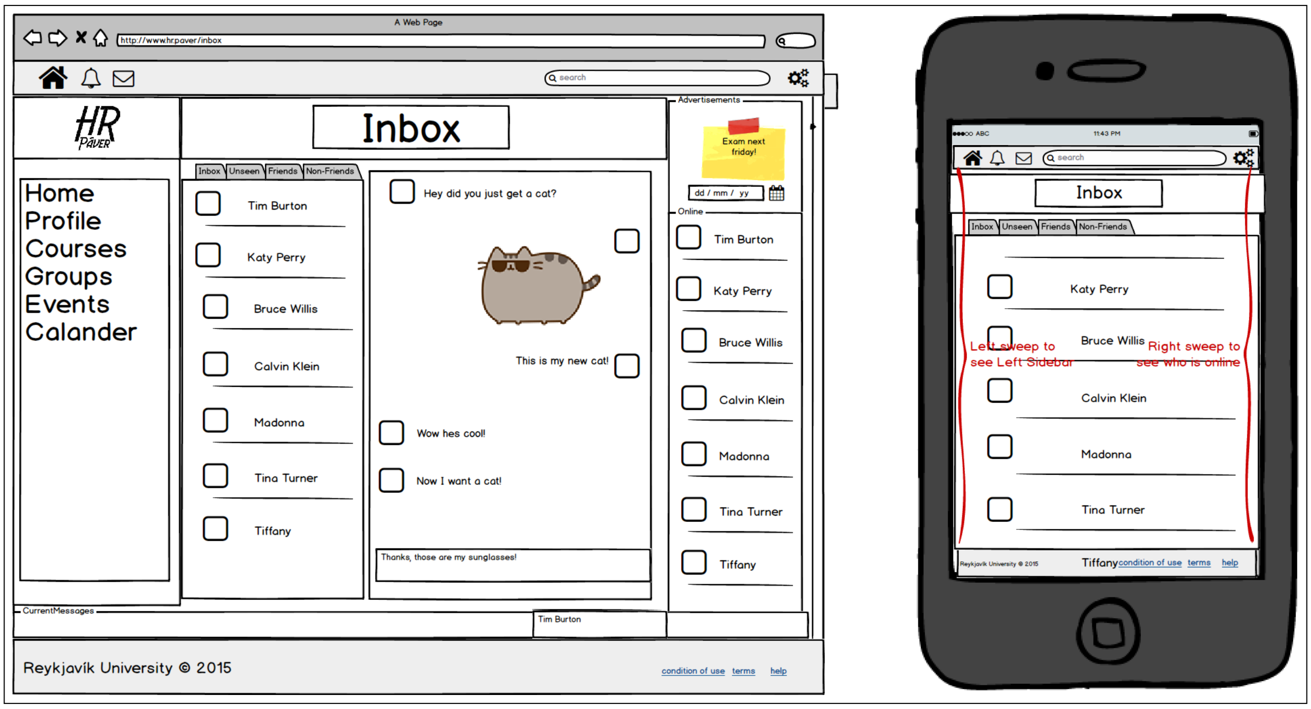
Recover Password

To recover a users password the user must input his profile's active email address.



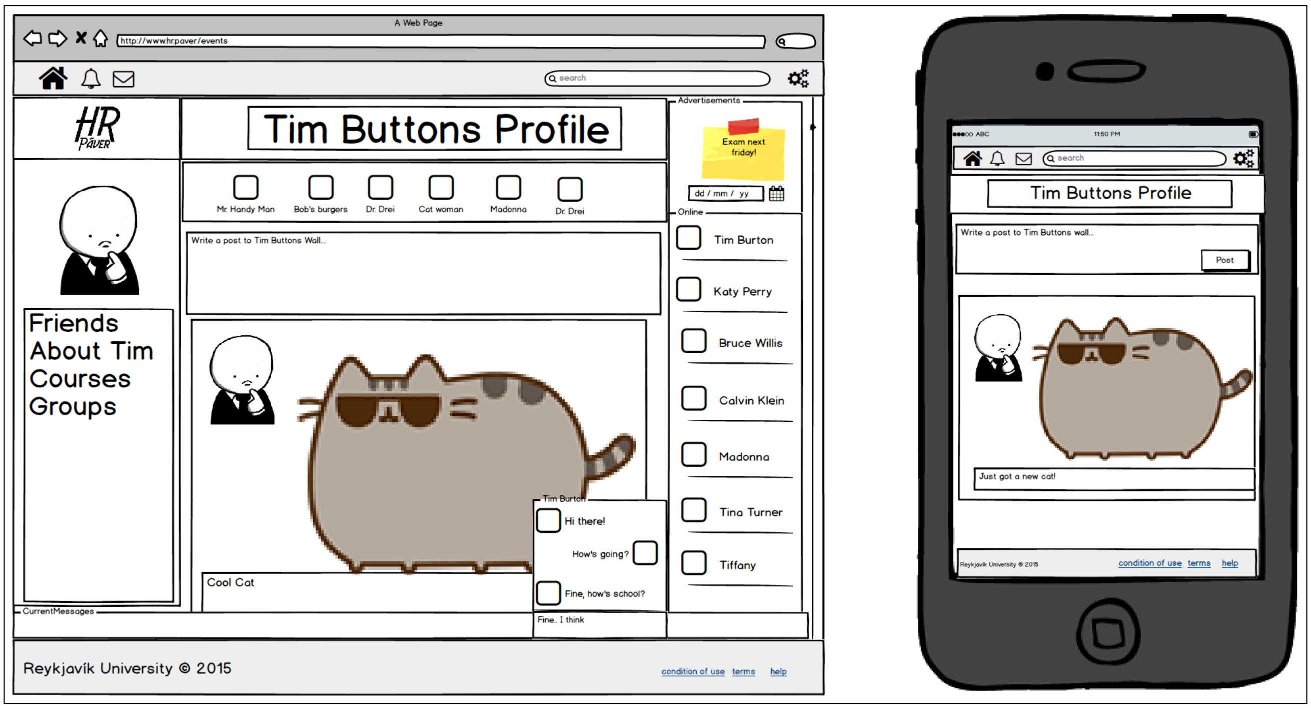
Home

The user's Home page shows the news feed from his friends. The page allows the user to post statuses or chat with friends. "Rolling active events" will show available events changing every few seconds. Header includes shortcut buttons: message cloud for inbox, a triangle for notifications and a wrench for settings.



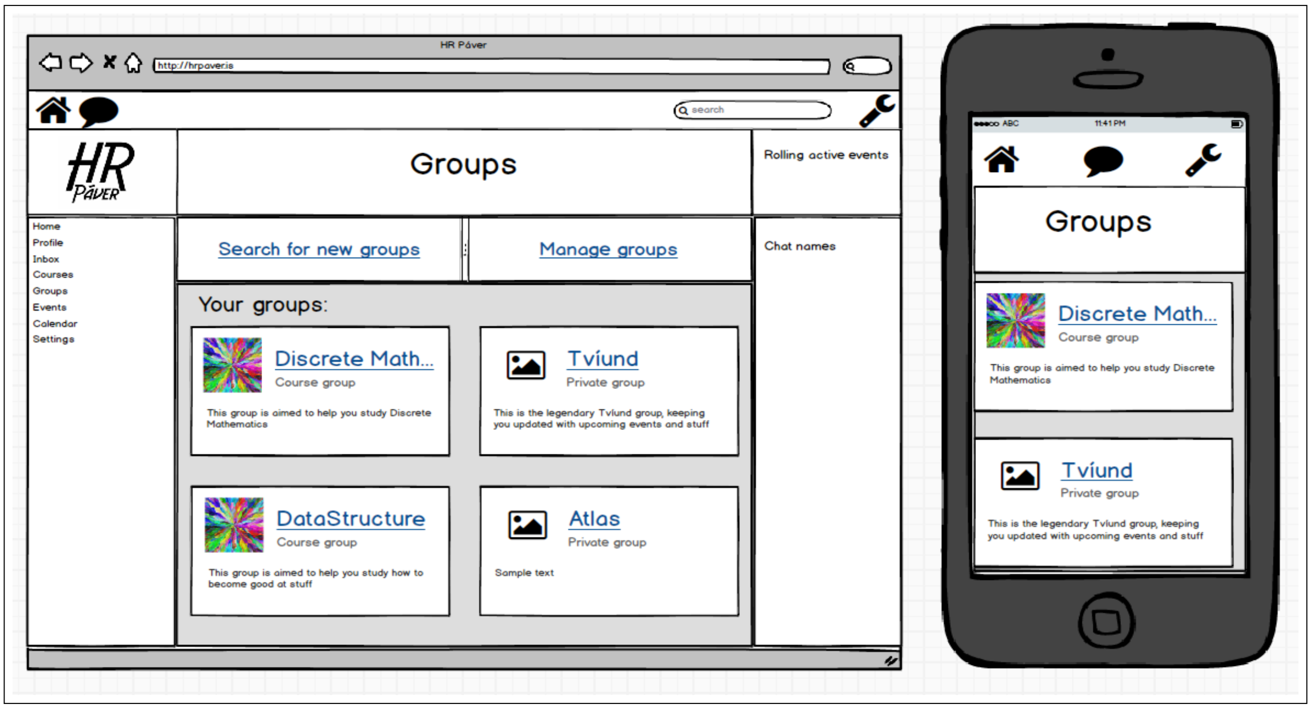
Inbox

Inbox shows overview of the messages the user has received or sent. He can search and filter his view by Friends, non-friends and unseen messages.



Profile

Users profile include a short bio about the users and a photo. Highlighted friends will appear above the news feed. Users can post on profile walls.



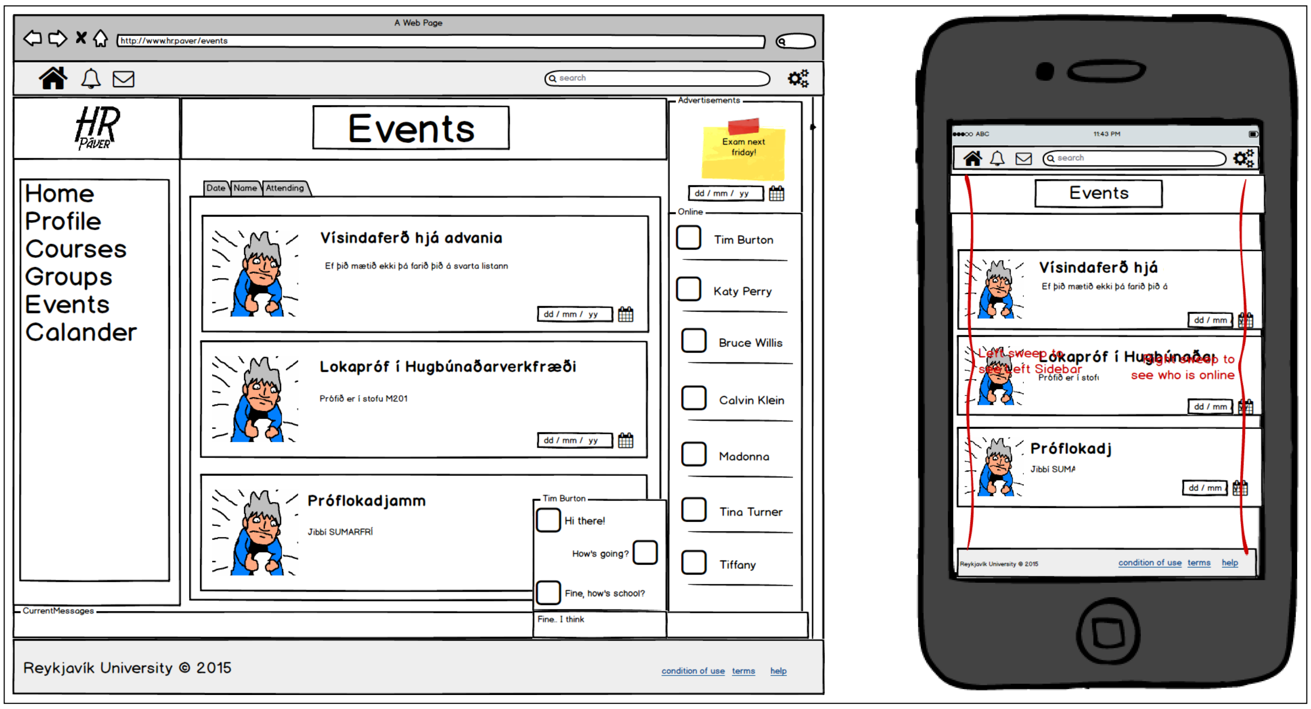
This group is aimed to help you study Discrete Mathematics

This is the legendary Tvíund group, keeping you updated with upcoming events and stuff

This group is aimed to help you study how to become good at stuff

Sample text

Groups
The group page shows the overview of the user's groups. He can search for new groups or manage his own and create or delete groups he owns.



Ef þið mætið ekki þá farið þið á svarta listann

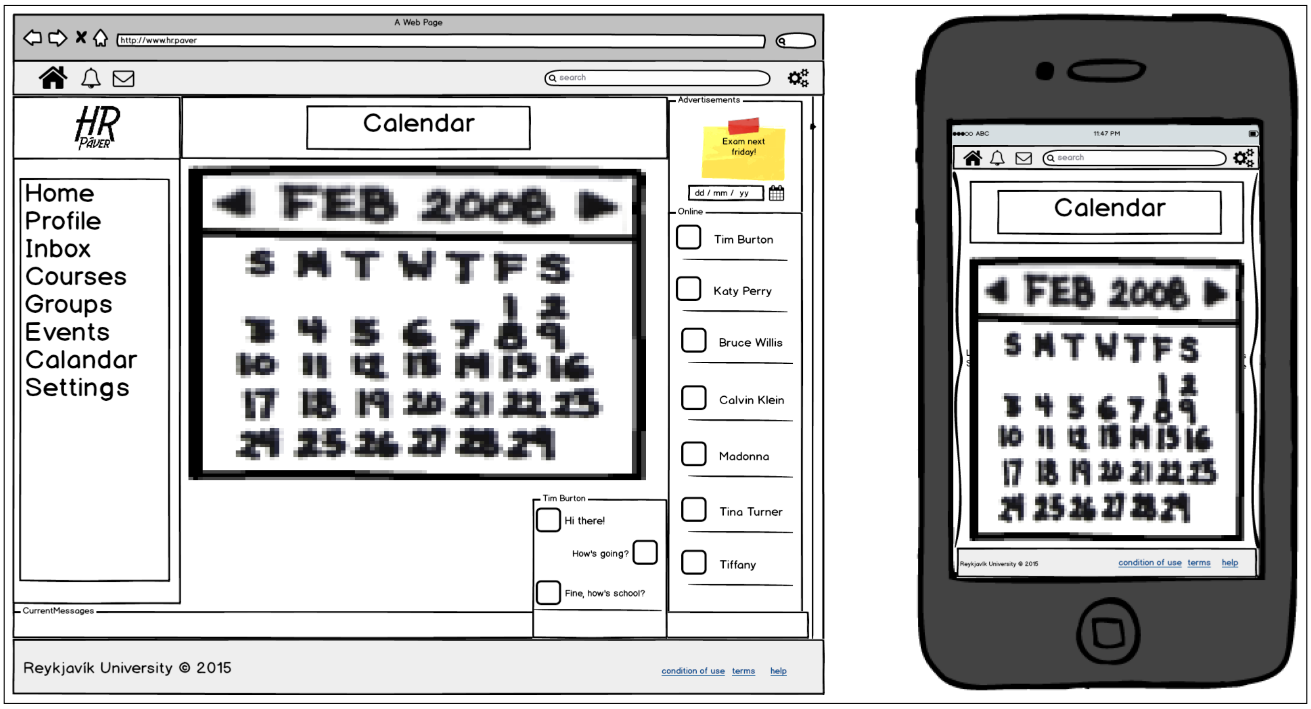
Prófið er í stofu M201

Jibbi SUMARFRÍ

Tim Burton
Hi there!
How's going?
Fine, how's school?
Fine. I think

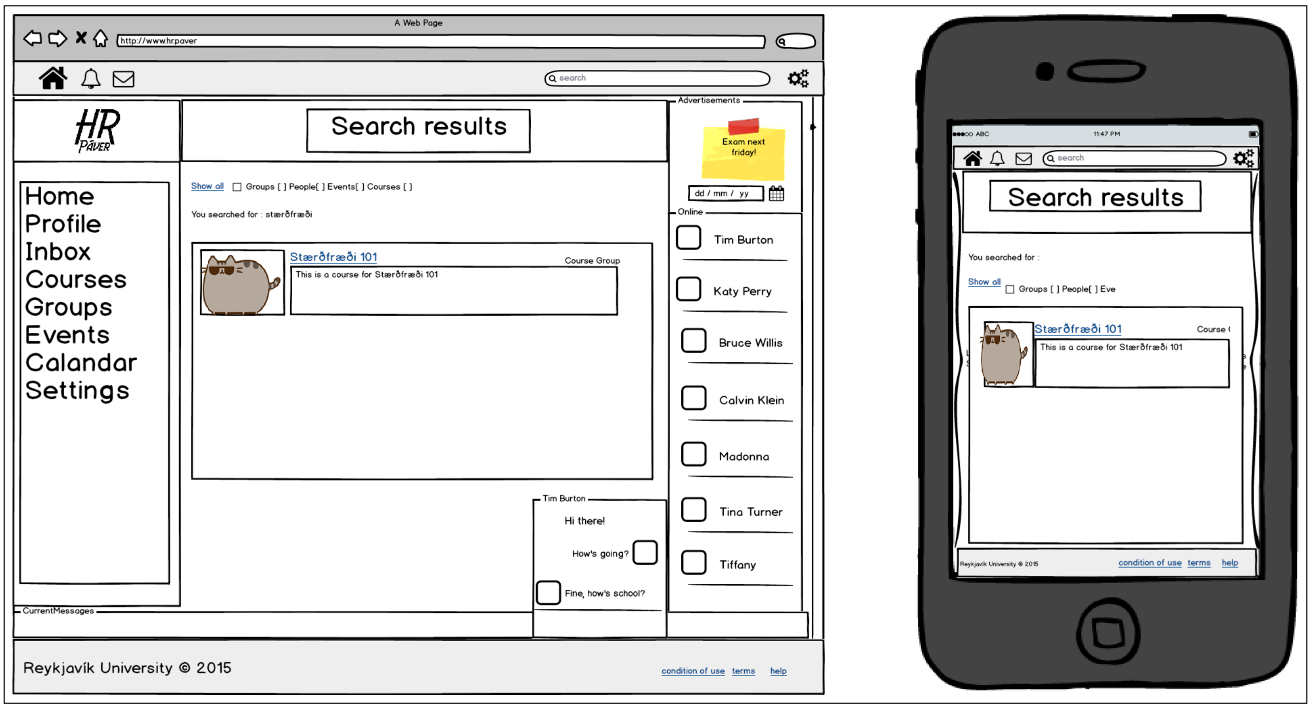
Reykjavík University © 2015

Events
The events page shows the available events and the user has the option to attend them.



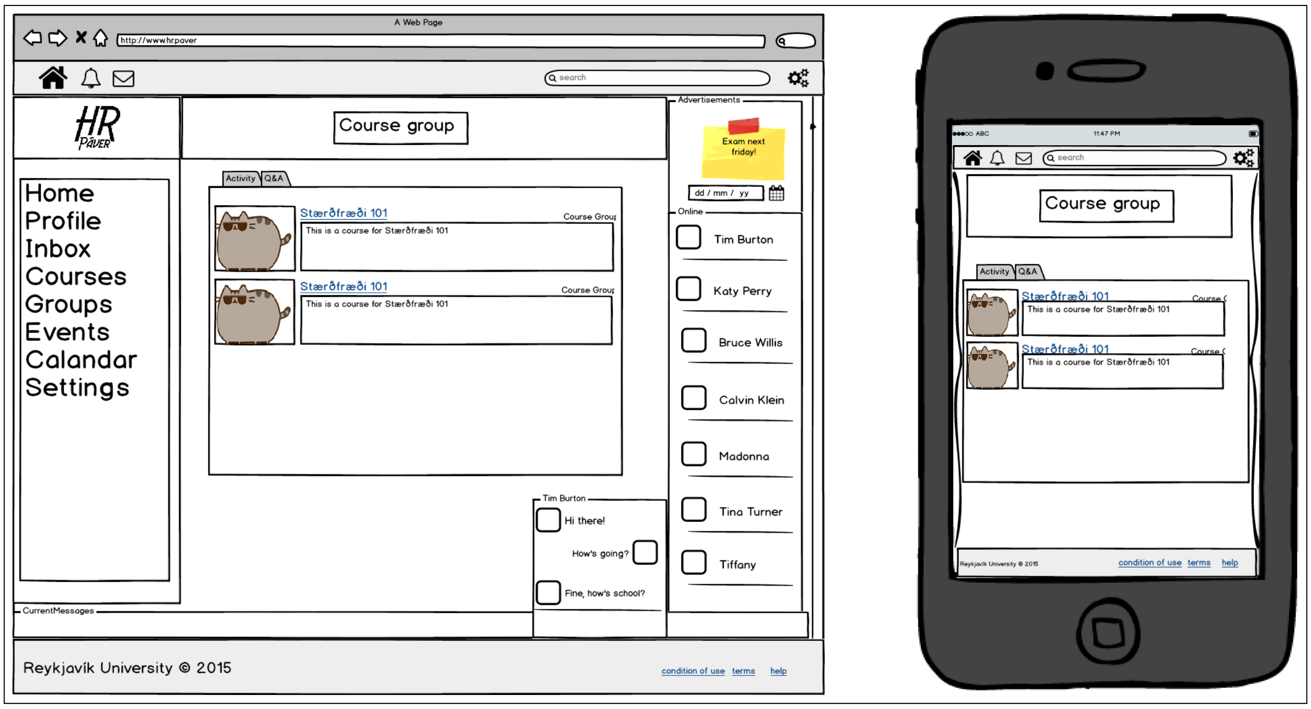
Calendar

The calendar page shows a calendar with all upcoming events that are available to the user.



Search Result

The search result page contains all users, visible groups and available events that match the user's search string.



CourseGroup

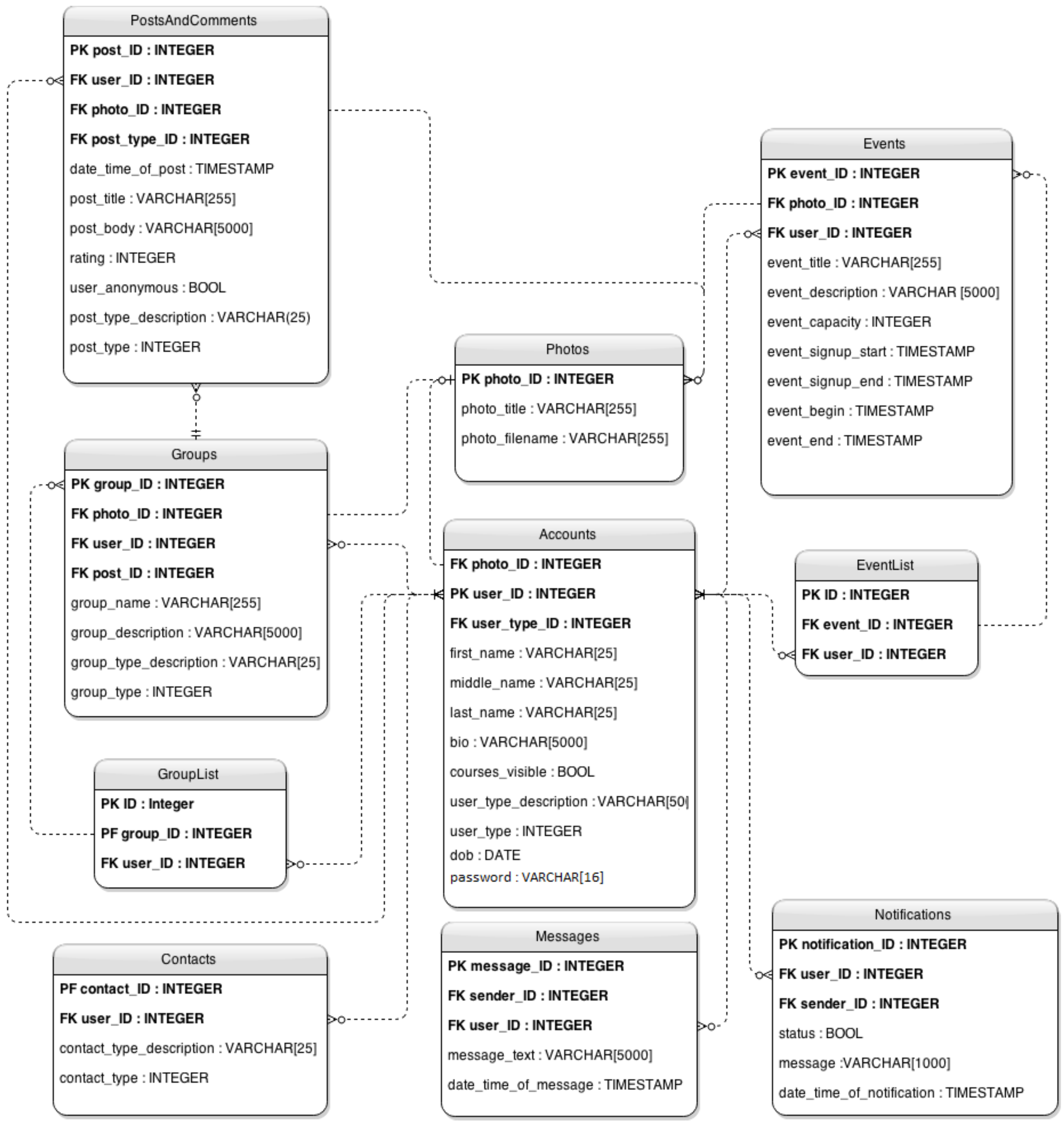
The course group page contains two walls, one for status updates and the other for Q & A regarding course material.

6 Database design

6.1 Database diagram

The diagram below provides a visual overview of the HR Páver database and the relations between its tables. The Table overview below includes additional details on the tables and columns.

Database diagram



6.2 Table overview

This section is the overview of all the tables that are in the HR Páver database. It is followed by specific information of what is in each table.

Table description

Table name	Description
Accounts	The list of accounts is maintained in the table Accounts. Each user has a type, there are six different types of users and each is represented with an integer value in the table. 1 is for Teachers, 2 is for Teaching assistants, 3 is for Students, 4 is for Users, 5 is for Other users and 6 is for Admin.
Groups	The list of groups is maintained in the table Groups. Each group has a type, there are four different types of groups and each is represented with an integer value in the table. 1 is for Course groups, 2 is for Private groups, 3 is for Closed groups and 4 is for Open groups.
GroupList	Each group has a list of users that is stored in the GroupList table.
PostsAndComments	The list of status posts, questions, answers and comments is stored in the table PostsAndComments. Each of these types are represented with an integer. 1 is for Status posts, 2 is for questions, 3 is for answers and 4 is for comments.
Notifications	The list of notifications a user has is stored in the table Notifications.
Messages	The list of messages a user has is stored in the Messages table.
Contacts	The list of contacts a user has is stored in the Contacts table. Each contact has a type, there are two different types of contacts and each is represented with an integer value in the table. 1 is for friend and 2 is for following.
Photos	The list of photos is stored in the Photos table.
Events	The list of events is stored in the Events table.
EventList	Each event has a list of users that is stored in the EventList table.

6.3 Table details

The following are the specific fields in each of the tables.

Accounts

Field	Type	Null	Key	Default	Extra
user_ID	Int		Primary		Auto increment
photo_ID	Int		Foreign	1	
first_name	Varchar(25)				
middle_name	Varchar(25)	Yes			
last_name	Varchar(25)				
bio	Varchar(5000)	Yes			
courses_visible	Bool			1	
dob	Timestamp				Date of birth
password	Varchar(16)				
user_type_description	Varchar(50)				
user_type	Int(4)				

Groups

Field	Type	Null	Key	Default	Extra
group_ID	Int		Primary		Auto increment
user_ID	Int		Foreign		
post_ID	Int		Foreign		
photo_ID	Int		Foreign	2	
group_name	Varchar(255)				
group_description	Varchar(5000)	Yes			
group_type_description	Varchar(25)				
group_type	Int(4)				

GroupList

Field	Type	Null	Key	Default	Extra
ID	Int		Primary		Auto increment
group_ID	Int		Foreign		
user_ID	Int		Foreign		

PostsAndComments

Field	Type	Null	Key	Default	Extra
post_ID	Int		Primary		Auto increment
user_ID	Int		Foreign		
photo_ID	Int		Foreign		
post_type_ID	Int		Foreign		
date_time_of_post	Timestamp				
post_title	Varchar(255)				
post_text	Varchar(5000)	Yes			
rating	Int			0	
user_anonymous	Bool			0	
post_type_description	Varchar(25)				
post_type	Int				

Notifications

Field	Type	Null	Key	Default	Extra
notification_ID	Int		Primary		Auto increment
user_ID	Int		Foreign		
sender_ID	Int		Foreign		
status	Bool			0	
message	Varchar(1000)				
date_time_of_notification	Timestamp				

Messages

Field	Type	Null	Key	Default	Extra
message_ID	Int		Primary		Auto increment
sender_ID	Int		Foreign		
user_ID	Int		Foreign		
message_text	Varchar(5000)				
date_time_of_message	Timestamp				

Contacts

Field	Type	Null	Key	Default	Extra
contact_ID	Int		Primary Foreign		Auto increment
user_ID	Int		Foreign		
contact_type_ID	Int		Foreign		
contact_type_description	Varchar(25)				
contact_type	Int(2)				

Photos

Field	Type	Null	Key	Default	Extra
photo_ID	Int		Primary		Auto increment
post_ID	Int	Yes	Foreign		
event_ID	Int	Yes	Foreign		
photo_title	Varchar(255)	Yes			
photo_filename	Varchar(255)				

Events

Field	Type	Null	Key	Default	Extra
event_ID	Int		Primary		Auto increment
user_ID	Int		Foreign		
photo_ID	Int		Foreign		
event_title	Varchar(255)				
event_description	Varchar(5000)				
event_capacity	Int				
event_signup_start	Timestamp				
event_signup_end	Timestamp				
event_begin	Timestamp				
event_end	Timestamp				

EventList

Field	Type	Null	Key	Default	Extra
ID	Int		Primary		Auto increment
event_ID	Int		Foreign		
user_ID	Int		Foreign		