Requirement and Analysis Document for Min5a

Naomi Espinosa, Robin Hallabro-Kokko, Theodor Khademi, Hanna Tärnåsen
& Johanna Wiberg

Date: 25/10
Version: 1

1 Introduction

No single student is perfect, everyone needs help in their time at university. A lot of information gets pumped into your brain every day and on top of that you need to know where your classes are, how much time you need for studying, when your exam is etc.. Unfortunately, a lot of this extra information is spread throughout many different providers, which makes it easier to forget and harder to keep track of. What our application hopes to do is create a central hub with all relevant information from their university, in this case Chalmers Institute of Technology. The application lets the user input the courses they are taking at the moment and gives the student all the information baked into one place. The application sets up the combined schedule of the courses, gives a countdown to the courses exam times and allows the student to make to-do lists and goals to keep track of what to do. The most prominent feature of the application however is the inclusion of a special timer which uses the Pomodoro technique, 25 minutes of studying followed by a 5 minute break. Research shows that study sessions are most effective in small short chunks as your brain is better at processing information at brief, repeated sessions.(1) Were hoping that through this application, stress and burden on students will decrease further.

1.1 Definitions, acronyms, and abbreviations

User story: A user story is a short description of a feature a user or customer wants. It is written in the perspective of the customer and starts with: "As a user I want...".

Pomodoro: Pomodoro is a studying technique that makes your work more effective. It's made up from a 25 minute studying interval and a five minute break.

Code coverage: Code coverage is a measurement on how many lines, methods or blocks is tested in an application. Meaning if every line in an application is tested the code coverage is 100%.

JavaDoc: JavaDoc is a documentation generator. You use it to document your code so other programmers can easily understand what you have written.

Build Automation: The process of automating the creation of a software build and the associated processes including: compiling computer source code into binary code, packaging binary code, and running automated test.

Continuous Integration (CI): A development practice that requires developers to integrate code into a shared repository.. Each check-in is then verified by an automated build, allowing teams to detect problems early.

Pushing/Pulling to Master: Commands that are user to upload/download remote changes from the to your own local project.

Branching: A system used to isolate work from the main project to try decrease the chance of harmful errors.

2 Requirements

.

2.1 User Stories

*Not implemented

1. As a Student, I want a timer that I can start and pause as I want, so that i can take quick brakes.
2. As a Student, I want to be able to see my study time live so I know how much time I have left and how far I have come.
3. As a Student, I want a HomePage with relevant information so I will feel more structured.
4. As a Student I want easy access to the main components that will be displayed static for me so that I can easily find where I want to go.
5. As a Student, I want to be able to add courses easily on the courses page so it lands in my list for easy access.
6. As a Student, I want to be able to edit my courses in case a spelled or did something wrong.
7. As a Student, I want to be able to end course when I'm done with it so it doesn't bother me any more.
8. As a Student, I want to be able to delete my courses if I happen to make a mistake or i for some reason don't want it.
9. As a Student, I want to be able to have the timer on a separate page so that i wont get as distracted.
10. As a Student, I want to have my data saved when I close the application.
11. As a Student, I want to be able to log into an account so that I can save my own info.
12. As a Student, I want to be able to edit my account if I have changed my mind.
13. As a Student I want to have an overview of a specific course so I can see all related course information in one place.
14. As a Student I want to be able to make deadlines so that I can have an overview of my studies.
15. As a Student, I want the possibility to make a "to-do-list" connected to courses so that I easily and fast work with quick short term tasks or open long term tasks
16. As a Student I would like to save all the important contact information of teachers and colleagues in a well sorted list
17. As a Student, I want to be able to see my teachers contact info on the course page so I don't have to search for it.

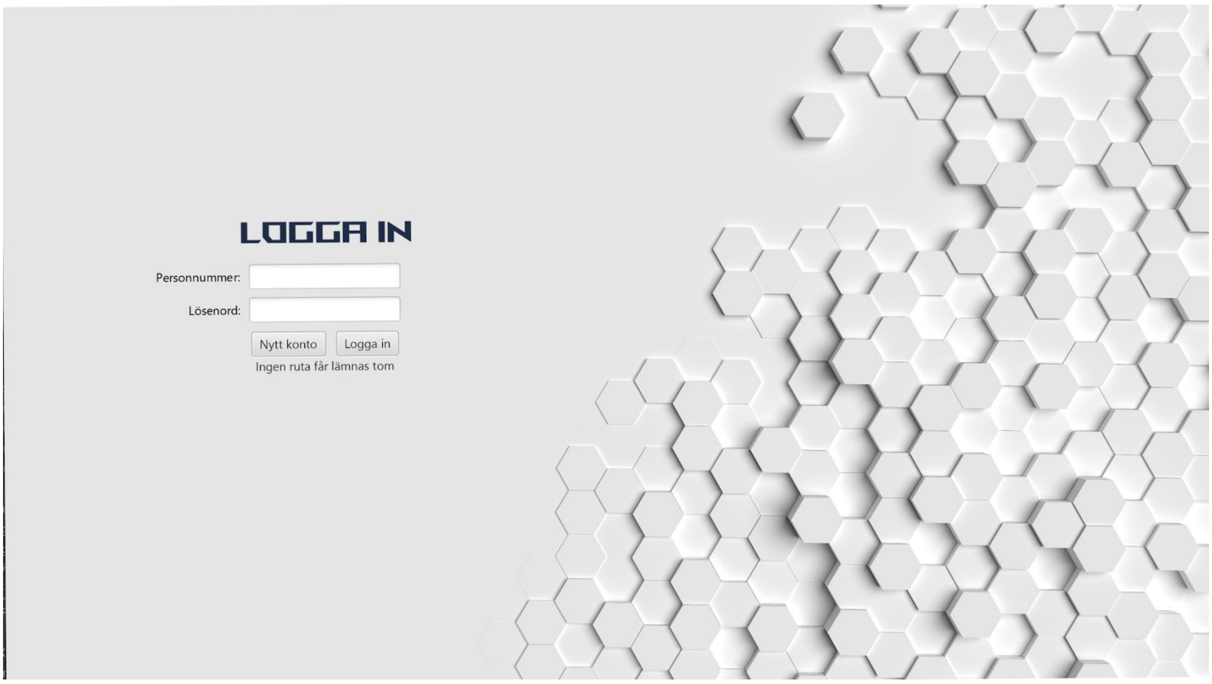
18. As a Student I want to be able to cancel an ongoing timer in case I have to interrupt my studies.
19. As a Student, I need a way of managing the time I spend in various courses to have a good overlook of my studies.
20. As a Student, I want to be able to set my own grade in the courses so that I can easily see the results of my goals and statistics and compare them.
21. As a student i want my timer to be able to minimize so that I can proceed to work with other things will the timer is running.*
22. As a student, I want my active courses to be sorted after time so that the most relevant course is being displayed for my comfort.*
23. As a student, I want to be able to show what courses are shown on the homePage by favoriting them so i can choose.*
24. As a student, I want to be able to have a settable countdown for the exam so that I will be able to plan better and have an overall better overview.*
25. As a student, I want to be able to write quick notes connected to each course so that i can come back later to evaluate further actions.*
26. As a user I want my scheduling data from TimeEdit to be visible directly in my courses inside the application.*
27. As a student, I want an editable dashboard so i can decide what information I find most relevant.*
28. As a Student I may want help focusing on my studies and remove distractions by locking down my computer.*
29. As a student I want to see today's lunch menus (fr. table d'hôte) from nearby restaurants directly inside the application.*

2.2 Definition of Done

- Before pushing to master, all functionality should be tested, reformatted and built by the Gradle build automation tool.
- If the user story concerns the model package it has to have 100% code coverage.
- The code should be well documented using javadoc.
- After pushing, the branch from the user story must first be accepted by the requirements that the CI Travis commands. Second, the branch should be reviewed and accepted by a member of the project group.

2.3 User Interface

When starting the application the first page the users faces is the login page. Here the user enters their personal number and password to log into an already existing account or create a new account.



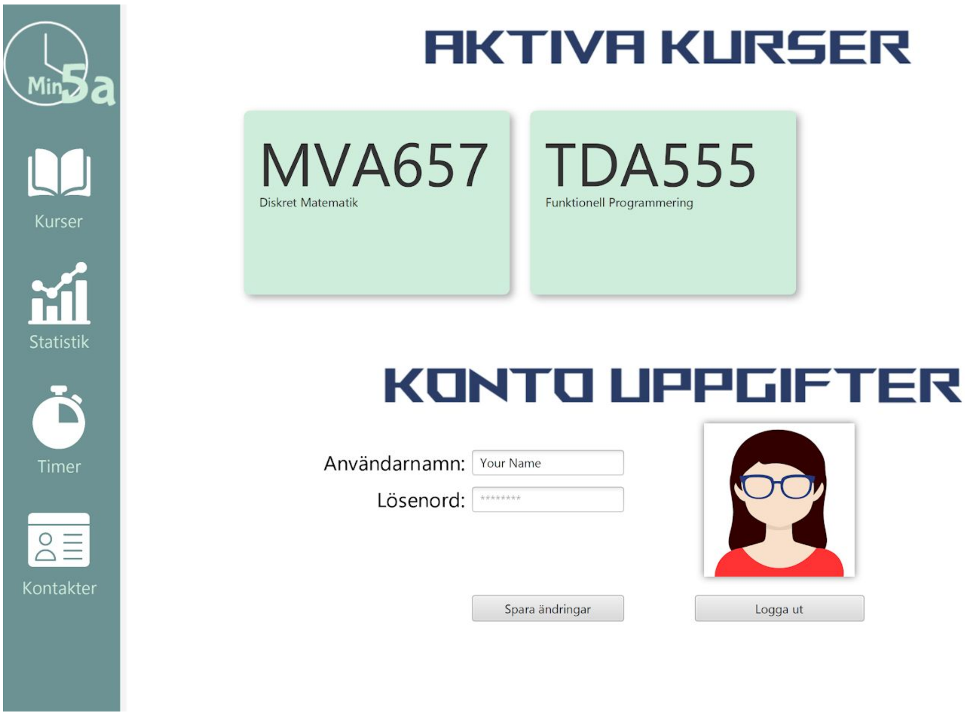
LOGGA IN

Personnummer:

Lösenord:

Ingen ruta får lämnas tom

When the user is logged in the first thing the user sees is the mainpage. Here they have easy access to their active courses and editing of account settings. You can also find a side panel to the left. This panel can be reached from every page of the application to make sure that they don't get lost.



AKTIVA KURSER

MVA657
Diskret Matematik

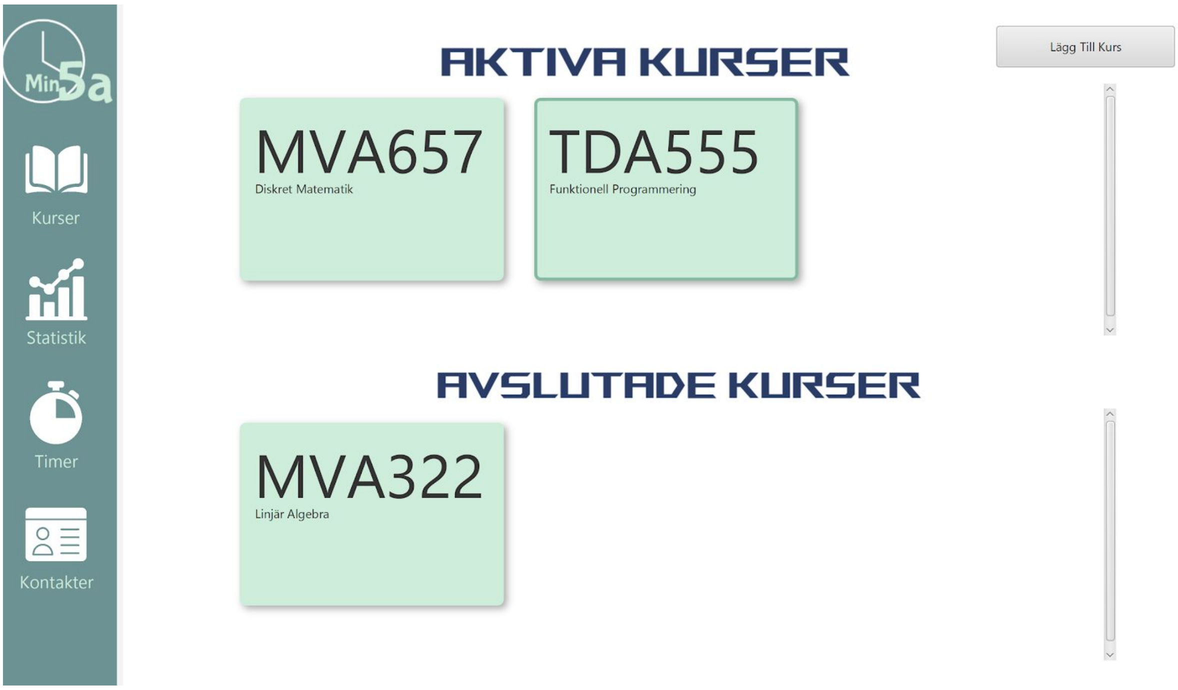
TDA555
Funktionell Programmering

KONTO UPPGIFTER

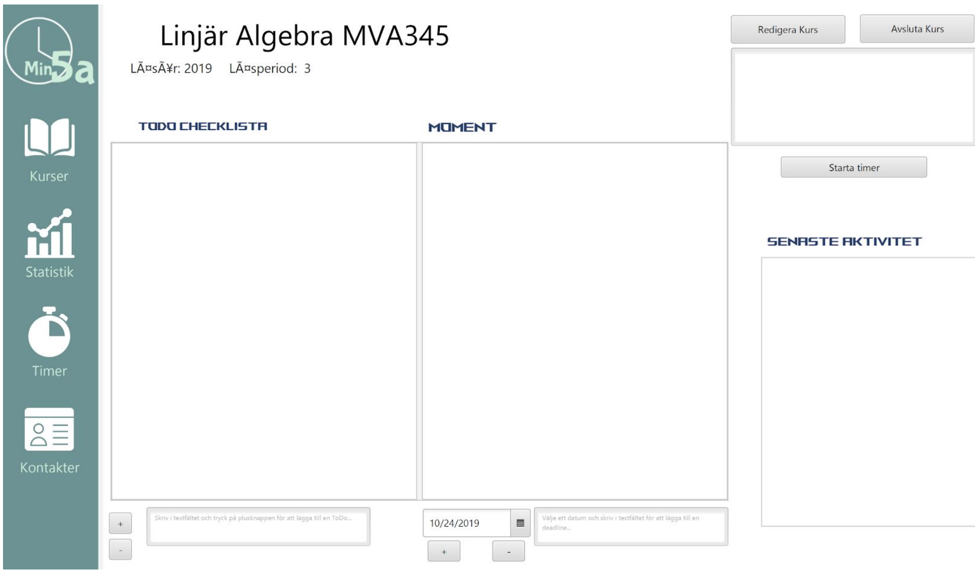
Användarnamn: Your Name

Lösenord: ********

To reach a courses page the user can click on one of the courses in the “aktiva kurser”-scrollpane or click on the “kurser” button in the side panel to get to the course selection page. Here they see a list of active courses and a list of inactive courses. On this page the user can also create a new course by clicking on the button in the top right corner. When they create a course they have to fill in the courses name, code, year it started and which period it is in. When a course is created they can find it on the course selection page and the mainpage.

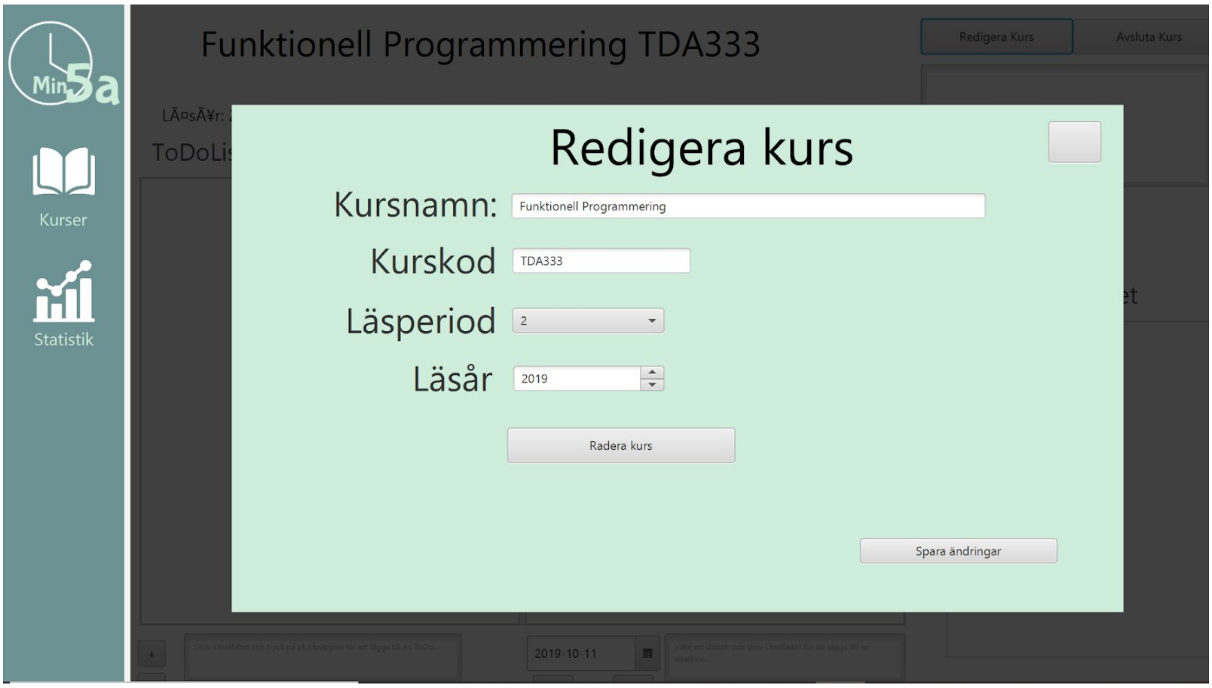


By clicking on a course the user navigates to that specific course page. Here you can create to-do's, keep track of your assignments and see your latest activities. From this page is it possible to edit the courses name, courses code, study period or study year in case something went wrong when you created it or some variable has changed.


Linjär Algebra MVA345

LÃ¤sÃ¥r: 2019 LÃ¤speriod: 3

10/24/2019
Min5a
Kurser
Statistik
Timer
Kontakter
Redigera Kurs
Avsluta Kurs
Starta timer
TODO CHECKLISTA
MOMENT
SENASTE AKTIVITET
+
-
Skriv i textfältet och tryck på plusknappen för att lägga till en ToDo...
Välje ett datum och skriv i textfältet för att lägga till en deadline...



Funktionell Programmering TDA333

Redigera kurs

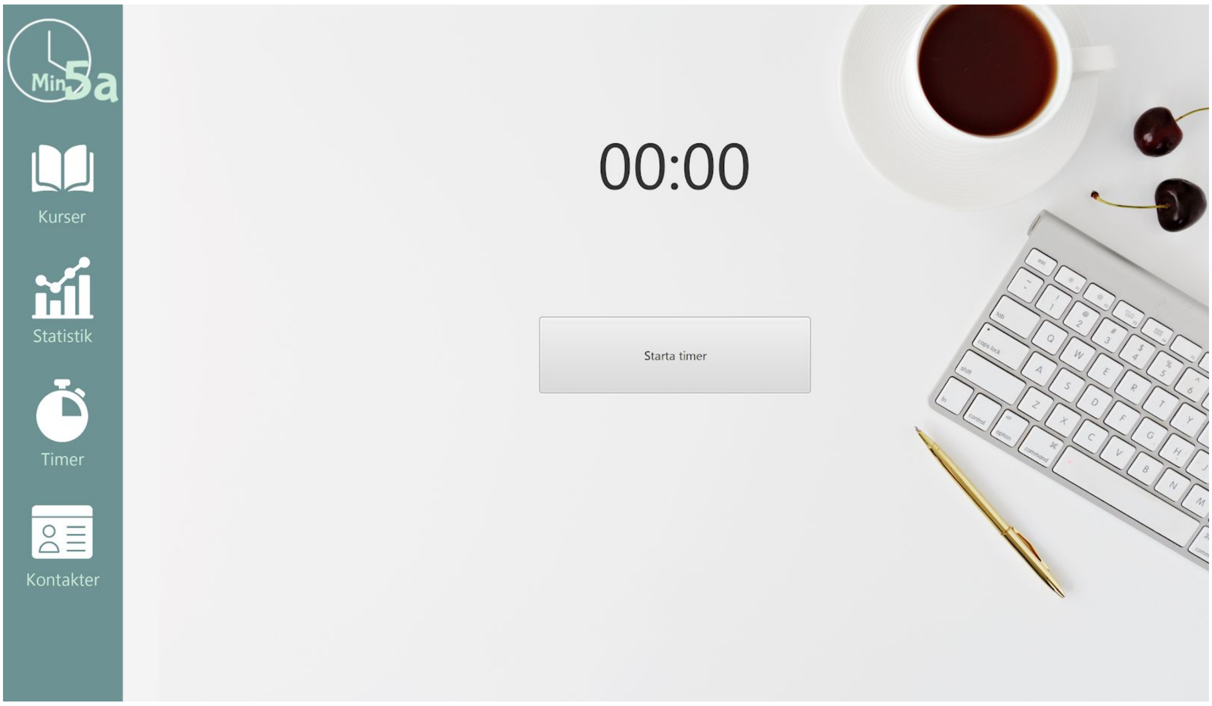
Kursnamn: Funktionell Programmering

Kurskod TDA333

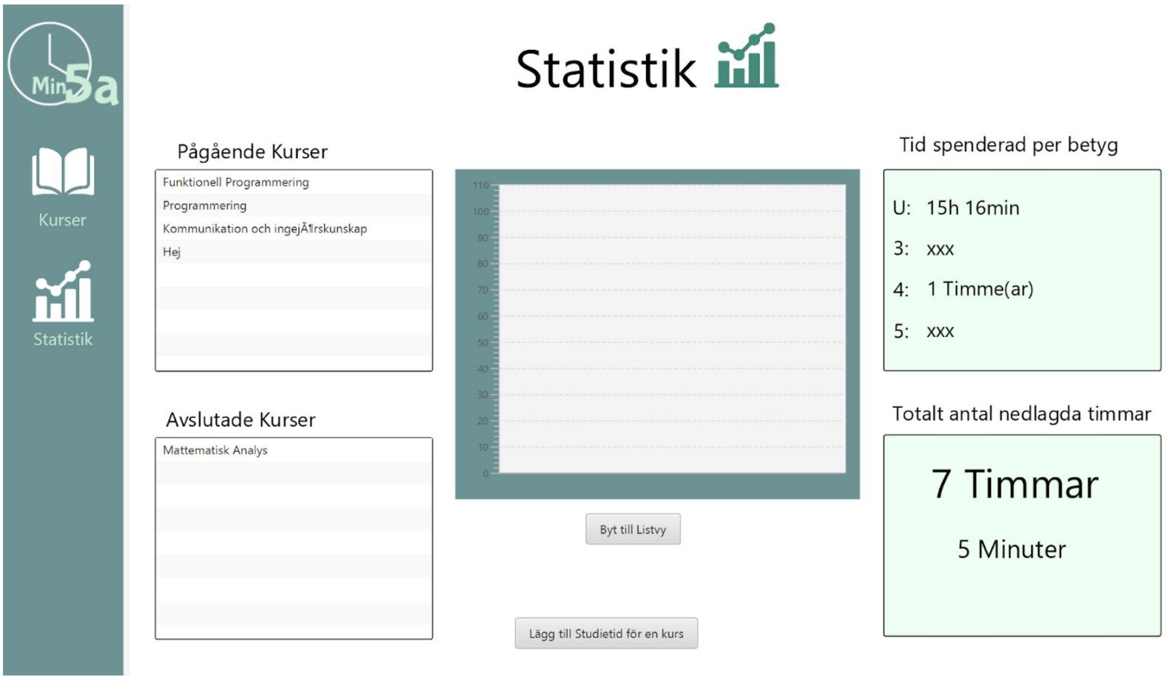
Läsperiod 2

Läsår 2019

Here the user can also start your timer by pressing the button “Starta timer”. When they do the user is redirected to a timer-page. Where the user can start a timer that counts up to 25 minutes, meant for studying, followed by a 5 minute break. It is possible to pause the timer.



By clicking on the “statistik”-button in the side panel you get to a page where all statistics is shown. Here you get an overview of how much you're studying to get a grade, how much you study overall, you can clearly see your ongoing and finished courses, and incase you missed to start your timer while studying you can add study time manually here.



Statistik

Pågående Kurser

Funktionell Programmering
Programmering
Kommunikation och ingejÃ¶rskunskap
Hej

Avslutade Kurser

Mattematisk Analys

Tid spenderad per betyg

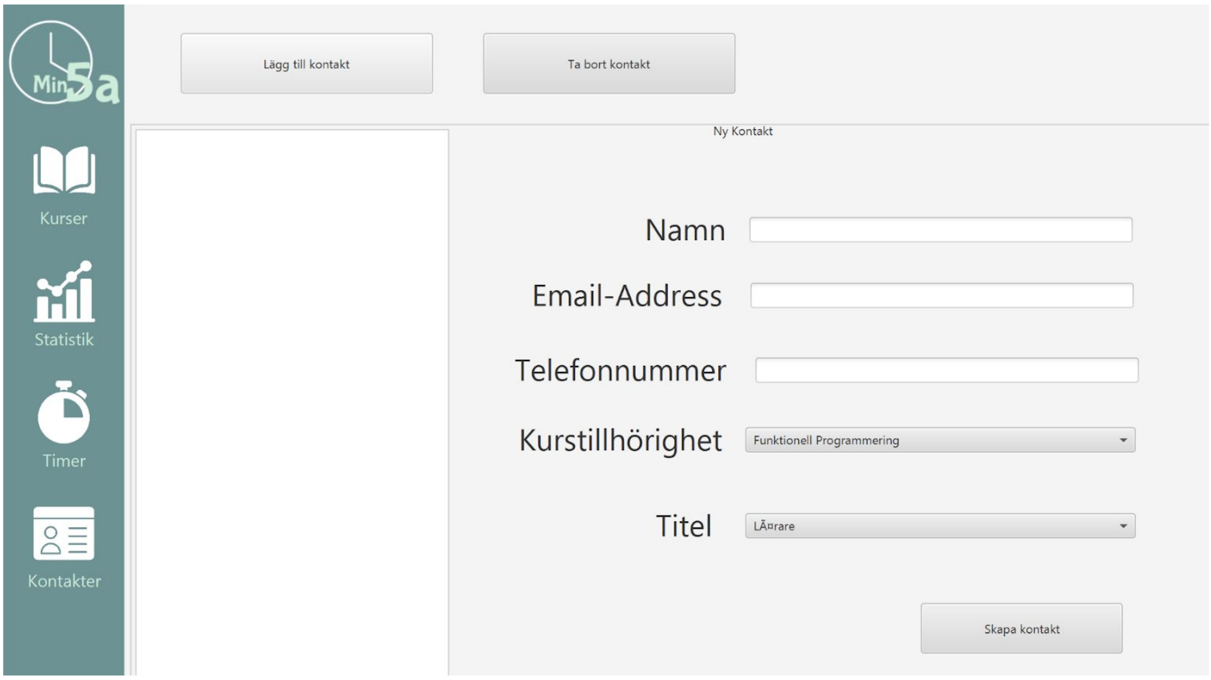
U:	15h 16min
3:	xxx
4:	1 Timme(ar)
5:	xxx

Totalt antal nedlagda timmar

7 Timmar

5 Minuter

By clicking on the “kontakter”-button in the side panel you get to a page where all contacts are gathered. When creating a contact you have to write in that contacts name, email, phone number, what course it belongs to and what title it has. When the contact is created it’s shown in a listview to the left.



Ny Kontakt

Namn

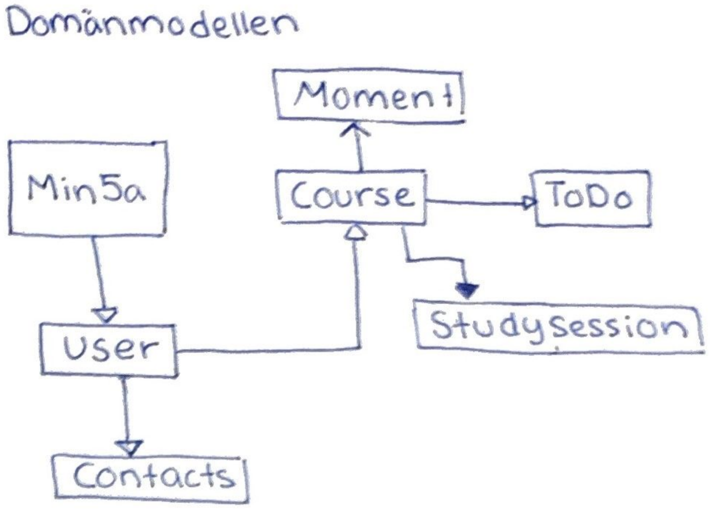
Email-Address

Telefonnummer

Kurstillhörighet: Funktionell Programmering

Titel: LÃ¤rare

3 Domain Model



3.1 Class Responsibilities

3.1.1 Model

Course
Course define what a course is and what a course can do. A course consists of a name, a course code, a study period and a year. A course also holds lists of course specific todos and moments.

Moment
Moment defines what a moment/assignment is and what it can do. A moment has an description and a date/deadline.

ToDo
ToDo defines what a todo item is and what it can do. An todo consists of a description.

Contact
The contact class defines what a contact is and what a contact can do. A contact consist of a name, an email address, a phone number and what course it's related to.

User

The user class defines what a user is and what it can do. The user consist of a name and an ID. The user has a collection of courses.

Timer
A class that represents the timer

StudySession
Represents one StudySession for a specific course.

3.1.2 View(fxml)

Contactspage
The visual for ContacsPage

CourseMainPage
The visual for CourseMainPage

CoursePanelItem
The visual for CoursePanelItem

CourseSelectionPage
The visual for CourseSelectionPage

HomePage
The visual for HomePage

MainPage
The visual for MainPage

StatisticPage
The visual for StatisticPage

TimerPage
The visual for TimerPage

3.1.3 ViewController

Contactspage
This is the whole “address-book” that is shown when you press the “kontakter”-button in the SidePaneln.

MainPage
This is the centre of the application and decides what page to be displayed and also handles the login functions

CoursePanelItem
This is the small boxes displaying courses that you can press on that are being shown in FlowPanes

CourseMainPage
This class represents a specific page dedicated to the specific course. It shows info about the course. It also allows the user to edit, complete and delete the course. The user can also add and delete moments and to-dos

HomePage
This is the frontpage that shows active courses and also info about the user

CourseSelectionPage
This page shows all of the users courses. Here the user can interact with the items to get to the courseMainPage. The user can also add new courses

PageLoader
A Factory Method that loads and creates all the fxml/Anchorpanes that we need

StatisticsPage
This page displays all the information about study-sessions and grades.

TimerPage
This is the class that represents the timer and functionality regarding controlling the fxml

PanelItemManager
Different classes share functionality while creating different Panel Items. This class displays panel Items in respective pane for respective class.

4 Bibliography

1. University of Illinois at Urbana-Champaign. "Brief diversions vastly improve focus, researchers find." ScienceDaily. ScienceDaily, 8 February 2011. <https://www.sciencedirect.com/science/article/pii/S0010027710002994?via%3Dihub>