Report

# Table of Contents

Table of Contents 2

Introduction 3

Problem Definition 3

Company Description 3

UP – Unified Process 4

Log – “Captains Log” 4

Inception 14

Organisation Type 14

SWOT 14

Business Case 15

Vision 15

Stakeholder Analysis 15

Feasibility Studies 19

Risk Assessment 22

Elaboration 23

FURPS+ 23

Use Cases 25

System Sequence Diagrams 36

Use Case Diagram 44

Domain Model 45

Sequence Diagram 46

Construction 47

Class Diagrams 47

Database 51

Entity-Relationship Diagram 51

Database Normalization Process 52

Transition 55

Reviews 55

Program User-Manual / AppAcademy Certificates Manager 55

aCerts – Prototype contra actual Program 58

OSCA 68

OSCA Threading 68

Summary 70

Bibliography 70

Appendices 70

# Introduction

## Problem Definition

Our customer, App Academy Aps, wants a program with following specifications:

* Standalone, single executable file.
* Generate a course certificate, which holds a course name, course participant's name and the date. This shall be saved as a jpg file.
* The certificates must have a unique IDs.
* The certificates shall be sent by e-mail with customizable text.

Furthermore, the client has wished for features which aren't essential for the program, but would be nice to have:

* “Drag 'n' Drop” functionality for generating a certificate template.
* Upload course material.

Due to close partnership with App Academy Aps we haven't been forced to set restrictions.

Our initial thoughts for solving this are to first focus on the design part, then hopefully construct the product without much difficulty. Also we have decided to describe App Academy Aps' business structure.

## Company Description

# UP – Unified Process

## Log – “Captains Log”

#### March 21st 2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

We had a meeting regarding our project for App Academy. During todays meeting we made a preliminary Entity Relation diagram draft and Class Diagram along with Domain Model diagram using Visual Paradigm v. 12.2. We had initial discussions regarding structure and design of the program and took a deeper look at Kristian’s draft regarding his wishes for the new platform.

We agreed to use GitHub as our version control, and decided to use English as our sole language throughout the project. Everything created or thought of will be digitalised during our project and stored online. “Paper” prototypes will be created in Balsamiq Mockups 3.

We’ve agreed that the next meeting will be held Tuesday 29th of March after class, due to Easter Holidays there will be quite a long time between this first meeting and the next, future meetings will be held twice a week.

We tried calling App Academy’s owner Kristian with enquiries regarding the design of the certificate itself. We’ve written him an email with our further questions.

Members Present: Dennis, Jonas, Emil, Fredrik. (All)

#### March 29th 2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

We’ve quickly looked through our basic java files. We’ve commenced our prototype. Besides this we’ve decided to add a toDo list for a quick overview of urgent tasks. Also there’s been added quickscribbles which is a collection of shorthand notes reg. Design decisions and debate/decisions in general. The prototype is quite vast and most GUI design decisions regarding admin user level have been decided and presented today.

For our next session we will be discussing secondary user level interface and prototyping it. We will also call Kristian and inform him about our Repo and give him the necessary information to hook onto it and be updated about our progress on the fly. We’ve decided that all software design aspects have to be done before any coding can commence.

Members Present: Dennis, Jonas, Emil, Fredrik. (All)

#### April 1st 2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Called Kristian, no reply but left a voicemail and hopefully he calls back. TableViews have been added to our prototype. Had a couple class members do short reviews on our prototype.

For our next session we will add full functionality to our prototype so that interaction will be possible and give a good idea on how the program is supposed to work.

Members Present: Dennis, Jonas, Emil, Fredrik. (All)

#### April 4th 2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Meeting was moved to April 5th 2016.

#### April 5th 2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Meeting was held and Kristian joined due to being at KEA. We continued working on our prototype and finished adding final functionality to it. Kristian had a look at our prototype and was satisfied with our design decisions and the prototype itself. We talked with him about functionality and we arrived at a decision to also add a certificate generator where he could edit the position of data printed along with certificate image. We have thus finished our first iteration of the program and it’s fully functional in balsamiq.

For our next meeting we will be working on our use cases and flow diagrams.

PS – We’ve decided to postpone our Friday meeting due to submissions and general studies.

Members Present: Dennis, Jonas, Emil, Fredrik (All) + Client.

#### April 11th 2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

We’ve met with Asger and had him review Kristian’s requirements as per email from 18.03.2016. He accepted the requirements and the project but added that statistics from database would be a good idea. Nice to have feature. Afterwards we worked on our activity diagram and added a casual Use Case. We’ve agreed that all software design will be done before April 23rd 2016, so that coding can commence after that date.

Next meeting: We will work further on software design in general. Specifics will be agreed upon then.

Members Present: Dennis, Jonas, Emil, Fredrik (All) + Asger. Feedback & open discussion.

#### April 12th 2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

A fully dressed use case for template generation has been made. Certain points regarding general design have been added to our qscribbles file. These were added during the making of our Use Case where we realised certain flaws in our design.

Next meeting: We will be looking at the assignments requirements from our teachers and client, having assessed those we will spend some time getting and overview from which we can plan our further approach.

Members Present: Jonas, Emil, Fredrik (Dennis had some KEA business, accepted by group beforehand).

#### April 18th 2016 / Iteration 1 has ended

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

We’ve made an initial project plan sketch. We’ve tried gathering an overview of missing tasks and things that need to be done for our project to be complete, so that we can assess our time and what’s required. We’ve made a crude priority list of what needs to be done first and agreed upon a fixed meeting schedule that requires more hours than we’ve put into the project thus far. From this point on we have 13 group sessions planned at 10 hrs. each. So a total of 520 man hours left before deadline, this is taking into account school classes that will be dedicated to our project, we haven’t taking these into account due to their vague nature.

Next meeting: Tomorrow during our next meeting our focus will be 100% directed towards ITO requirements and completing as many of them as possible.

Members Present: Jonas, Emil, Dennis, Fredrik (All).

#### April 19th 2016 / Iteration 2 has begun

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Finally managed to get started on our Report. We’ve completed our feasibility studies along with Stakeholder analysis’. This also means that due to our feasibility studies that we now know the exact amount of time that we have left to complete the entire project and will be adding a countdown on developer hours in the captains log from here on.

Next meeting: ITO(Risk Analysis), Commence description of UP in Project.

Members Present: Jonas, Emil, Dennis, Fredrik (All).

Developer Hours: 10 hours on ITO.

ITO: 10/20 – SWD: 0/70 – OSCA: 0/10 – SWK: 0/30 – Buffer 0/61.5

ETA: 181,5 Hrs Remaining pr. Developer

#### April 22nd 2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Today we had some trouble finding a place to study as KEA was closed for the holidays. However we managed to get into KU, but two hours after schedulded meeting time. We finished our risk analysis from ITO. We tried to proceed with documenting the fact that we’ve used UP while working on the project, but we figured that we need to consult Asger about this as we don’t know how to do it properly. Instead we initiated a mapping of the tables in our database along with normalization process while the teachings from class are still fresh in memory.

Next meeting: Ask Signe about Normalization, Ask Asger about UP, Ask Kristian about SWOT

Continue DB Normalization, Finish UP, Finish ITO SWOT

Members Present: Jonas, Fredrik (2 Short).

Developer Hours: 2 hours ITO, 2 hours from budget in unforseen closing of KEA, 6 hours SWK

ITO: 12/20 – SWD: 0/70 – OSCA: 0/10 – SWK: 6/30 – Buffer 2/61.5

ETA: 171,5 Hrs Remaining pr. Developer

#### April 26th 2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

We’ve finally finished normalization of our database and initial scripts for the database. We’re going to implement foreign keys next time. This log was written on April 29th due to a glitch in the matrix. We managed to get in touch with Signe as well regarding questions for normalization, and got some good feedback on our process.

Next meeting: Ask Asger about UP, Ask Kristian about SWOT

Finish UP, Finish ITO SWOT

Members Present: Dennis, Fredrik, Jonas&Emil were there for half a session.

Developer Hours: 7 hours SWK, 3 hours SWD

ITO: 12/20 – SWD: 3/70 – OSCA: 0/10 – SWK: 13/30 – Buffer 2/61.5

ETA: 161,5 Hrs Remaining pr. Developer

#### April 29th 2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

We completed our Database script with all the details and added the dummy information from our normalization processes. We revisited some old use cases and our now prototype version 1 has been amended to fit the current level of development. We’ve added certain features that had been forgotten and removed unnecessary fluff. We also found the necessary code and information to implement FTP and SMTP communication into our program and got that working with our server, so now we will be able to store CourseMaterial and Certificate copies online and send out certificates using emails on course participants from the database. A very productive day indeed.

Next meeting:

Add FTP SMTP editor to Prototype V2

Ask Asger about UP

Ask Kristian about SWOT

Finish UP

Finish ITO SWOT

Members Present: Dennis, Fredrik, Jonas.

Developer Hours: 3 hours SWK, 7 hours SWD

ITO: 12/20 – SWD: 10/70 – OSCA: 0/10 – SWK: 16/30 – Buffer 2/61.5

ETA: 151,5 Hrs Remaining pr. Developer

#### May 3rd 2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Today we identified processes according to their correct UP position. We finalised all our Use Cases and the Domain Model. We started Use Case Diagrams, but we havn’t been able to finish this during todays session. FURPS+ has been finalised. Vision has been commenced, needs further inspection though. We’ve agreed for a conference call with Kristian on May 10th for a short review and key information for our SWOT analysis. We’ve restructured our project folder to reflect our progress and iterations in a more visible manner. We’ve also been forced to send Emil a short message reg. his performance and enthusiasm as of late. He’s not been attending group sessions as agreed and when present he doesn’t participate and leaves halfway through. We hope for improvement.

Next meeting: Finish SWD

Revise Vision

Members Present: Dennis, Fredrik, Jonas, Emil 1/3 Session

Developer Hours: 10 hours SWD

ITO: 12/20 – SWD: 20/70 – OSCA: 0/10 – SWK: 16/30 – Buffer 2/61.5

ETA: 141,5 Hrs Remaining pr. Developer

#### May 6th 2016 / Iteration 2 has ended

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

We've identified several issues with our UP. We need to iterate more frequently. Be better at pin pointing iteration points. We've restructured our approach to be more focused on iterations in general. We've commenced coding GUI features and finalised a 3rd iteration of our prototype. We've finished system sequence diagrams and prepared a general report document. Our approach thus far in general has been too heavily focused on on an approach that risks being classified as Waterfall model and iterations will regain focus now.

Next meeting: See Progress Table

Finalise preliminary code

Members Present: Dennis, Fredrik, Jonas, Emil

Developer Hours: 7 hours SWD, 2 hours ITO, 1 hour SWK

ITO: 14/20 – SWD: 27/70 – OSCA: 0/10 – SWK: 17/30 – Buffer 2/61.5

ETA: 131,5 Hrs Remaining pr. Developer

#### May 10th 2016 / Iteration 3 has begun

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Group met as planned, we adjusted and corrected Use Cases to fit the current stage of our program and added 'View Course Participant' use case. Prototype has changed yet again and will have to be further refined at next meeting. We also finished a very thourough Class Diagram using MVC(Model, View, Control) and also implemented Logical Architecture Diagram, which still needs some work. We've decided that if any new variable/functions/methods etc. need to be implented into the code which will be written down using Class Diagrams next time, we will adjust current Iteration 3 Class Diagrams according to these minor changes.

Kristian Conference Call - Never Called as agreed.

Next meeting: See Progress Section

Correct Prototype

Finish Class Diagram, Model part

Finish remains on Logical Architecture

Try Calling Kristian

Commence actual code using Class Diagrams.

Members Present: Dennis, Fredrik, Jonas, Emil

Developer Hours: 2 hours SWD, 8 hours SWK

ITO: 14/20 – SWD: 29/70 – OSCA: 0/10 – SWK: 25/30 – Buffer 2/61.5

ETA: 121,5 Hrs Remaining pr. Developer

#### May 13th 2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Today we had our regular friday crunch day. We finalised our Logical Architecture and finished Model part of our Class Diagram, we've finally been able to commence some real coding and have almost finished our GUI part of aCerts. It's going to be great fun next time, where we will continue coding. We found a minor oversight, it's actually impossible to assing our Course Participants to a Course, this will be corrected for next time. We still havn't been able to get a hold of Kristian which is very frustrating since we need his help with certain parts of our ITO assignment, SWOT analysis etc. Well, maybe next time.

Another thing we can observe now is that we've budgetted our time incorrectly, We've exceeded our SWK total Budget of 30 hours by 2 and we're far from done. On the other hand we have alot more time assigned to SWD than will be used with concurrenct ad-hoc tasks. We will agree on how much time should be taken from Buffer and maybe some from SWD aswell during the next Metting, so that we can achieve a more accurate target.

Next meeting: See Progress Section

Correct Prototype / Add Course Participant Part

Try Calling Kristian

Continue coding GUI part of aCerts

Members Present: Dennis, Fredrik, Jonas, Emil

Developer Hours: 3 hours SWD, 7 hours SWK

ITO: 14/20 – SWD: 32/70 – OSCA: 0/10 – SWK: 32/30 – Buffer 2/61.5

ETA: 111,5 Hrs Remaining pr. Developer

#### May 17th 2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Today has been a primarily production day. We've been coding the entire GUI/View part of aCerts, we've only got some minor bugs to fix. There is not much to say as everything has been smooth sailing. Next time we will be implementing functionality and finalising Iteration 3.

Next meeting: See Progress Section

Try Calling Kristian

Add functionality to aCerts

Members Present: Dennis, Fredrik, Jonas, Emil

Developer Hours: 1 hour SWD, 9 hours SWK

ITO: 14/20 – SWD: 33/70 – OSCA: 0/10 – SWK: 41/30 – Buffer 2/61.5

ETA: 101,5 Hrs Remaining pr. Developer

#### May 20th 2016 / Iteration 3 ends

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Today has been the end of Iteration 3. We've implemented alot of functionality, everything has been fully commented and there is full interaction between our database and aCerts. We've also uncovered that MySQL 5.7.12 released 11th of april is incompatible with our Version 5.5.49. Our Syntax is working with 5.7.11 aswell, but this is only confirmed on a windows platform.

Next meeting: See Progress

Still need to get a hold of Kristian

Continue adding functionality to aCerts

Update Prototype

Update Class Diagram

Sequence Diagrams

Members Present: Dennis, Fredrik, Jonas, Emil(½ Session)

Developer Hours: 10 hours SWK

ITO: 14/20 – SWD: 33/70 – OSCA: 0/10 – SWK: 51/30 – Buffer 2/61.5

ETA: 91,5 Hrs Remaining pr. Developer

#### May 24th 2016 / Iteration 4 Begins / Transition Phase is main focus

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Today we continued refining our code by adding features such as being able to save certificates, add participants,

and some other small refinements such as adding FTP libraries and fixing the 'MySQL Settings' class. However

we are starting to feel the productivity level is falling, as we have now been working on the project for quite some time.

We will have to tackle this problem by keeping spirits high here in the final part of the work, as the hand in date

is now but a week away.

Next time we will aim to finalize the code by adding the final refinements and comments to the code.

We hope to do this as we can then start adding all our work to the report and thereby start finishing the project as a whole.

Members Present: Dennis, Fredrik, Jonas, Emil

Developer Hours: 6 hours SWK, 4 hours OSCA

ITO: 14/20 – SWD: 33/70 – OSCA: 4/10 – SWK: 57/30 – Buffer 2/61.5

ETA: 81,5 Hrs Remaining pr. Developer

#### May 27th 2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Our code is now finished to such an extent that we can show our client Kristian the result.

Therefore, we have agreed to do a skype meeting with him where we will screen share as we try out the program with instructions from Kristian. We have also revisited our class diagram for the last time to update it with the small refinements made to the code. Also, we have made a user manual for the program which explains how the program should be set up the first time when it is launched as well as a description of it's general usage. Finally, we have solved the OSCA assignment and thus finished its coding part.

Next time we will have the skype meeting with Kristian and start finalizing the report and the project as a whole.

Members Present: Dennis, Fredrik, Jonas, Emil

Developer Hours: 3 hours ITO, 3 hours OSCA, 2 hours SWK, 2 hours Buffer

ITO: 17/20 – SWD: 33/70 – OSCA: 6/10 – SWK: 59/30 – Buffer 4/61.5

ETA: 71,5 Hrs Remaining pr. Developer

#### May 31st 2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Inception

### Organisation Type

### SWOT

|  |  |
| --- | --- |
| Strengths | Weaknesses |
| Hvilke fordele har AppAcademy?  Vi har anciennitet, masser af erfaring og cases at gå ud fra. Det er en fordel iforhold til salg også. Vi har en god kombination at IT faglighed og egentlige uddannede lærere.  Hvor differentierer AppAcademy sig fra konkurrenter?  Der er 2 forskellige konkurrent typer. Fritidsundervisning, AppAcademy er indenfor knotortid. Indenfor skoleområdet er der manger der burger det som bijob, AppAcademy gør det fuldtid og er derfor mere professionelle I deres tilgang.  AppAcademys økonomi, konsulenter osv.?  Selvfinansieret, ingen banklån eller investorer. Det gør at I er fleksible og beslutninger kan tages hurtigt.  Kristian har en mentor tilknyttet, en voksen herre med masser af erfaring. | Hvad kunne AppAcademy gøre bedre?  Markedsføring, meget tid på salg. ROI kunne forbedres på marketings budget. Tiden der bruges på at få en ordre I hus er for høj.  Hvad skal AppAcademy undgå?  Det svære er at allokere resourcer til et nyt kursus, så der er nok material til at præsentere det for kunder. Farligt at man gætter forkert og kunderne ikke vil have det. Kvalitetssikring. Bagsiden ved at være lille kontra stor.  Hvilke faktorer udløser tab af afholdte kurser?  Dårlige undervisere. |
| Opportunities | Threats |
| Hvilke muligheder kan AppAcademy se I fremtiden?  Der er mange, på skoleområdet er der en trend med at IT og programmering er på fremmarch, AppAcademy ligger rigtig lunt I svinget til at opkvalificere disse folkeskole lærere til at komme op på niveau. Regering er igang med at forhandle gymnasia reform, I udkast står der at alle gymnasia studerende skal have en eller anden form programmeirng, her kan appacademy også byde ind.  Hvilke trends gør sig gœldende indenfor markedet pt?  Flere online video kurser. De store konkurrenter, har brugt mange resource på kursus lokaler, det er ikke længere en fordel for dem med den online trend der er for tiden. AppAcademy har ikke nogen lokaler men har brugt pengene på server infrastruktur til at kunne tilbyde online kurser istedet.  Hvilke teknologiske innovationer vil gavne AppAcademy? | Hvilke udfordringer står AppAcademy overfor?  Den Største trussel på skolesiden. Center for UNdervisningsmidler forkortet CFU. DEt er offentlige insitutioner som skal være et slags bibliotek for skolerne, de overvejer at tilbyde kurser eftersom at bibliotek ikke er I efterspørgsel, de kan derfor tlbyde kurser billigere da de har allokeret resource til medarbejder osv. De behøver ikek konkurrere på markedsvilkår.  De amerikanske international kurser er en klar trussel. Lynda, Coursera f.eks. Lokalt er Video kæmpe opportunity, Internationalt trussel.  Hvad gør AppAcademies konkurrenter der kunne indlfyde vores position?  Er den teknologiske udvikling med f.eks. e-kurser en trussel?  Er der kaptial problemer?  Man er sårbar som lille virksomhed I forhold til personale, der er ikke capital til at fortsætte hvis der er en lang down periode på salgs delen.  Er virksomheden sårbar pga. Størrelsen? |

### Business Case

### 

### Vision

This program shall function as a way of creating and sending certificates for course participants. The administrative course responsible for the program can add new information regarding courses and their participants.

### Stakeholder Analysis

**Stakeholders:** KEA, Testers (Class Mates), Project Group, App Academy(Kristian), Course Responsibles

PLACEHOLDER FOR FIG. 12.7 I ORGANISATION 5. UDGAVE

Grey Eminence: KEA

Hostage/ResourceStakeholder: Kristian

Hostage: Course Responsibles

ResourceStakeholder: Project Group

External Stakeholder: Testers

#### Grey Eminence: KEA

KEA as a Stakeholder:

We’ve decided that KEA has the role of a Grey Eminence. We’re not going to pay a lot of attention to KEA as an institution during our project, but it has the power to change the entirety of the project if it wants and sees an interest in it.

KEA’s attitude towards the project:

KEA has no direct wish for the actual program to be completed, but wishes the project group to succeed as the project group is part of their education system.

KEA’s potential for conflict:

KEA has accepted the terms of the project as an exam project and does overruled any saying Kristian might have regarding the continued direction of the project.

KEA’s resources:

They can provide help with all aspects as they provide teachings in all subject components needed to deploy the system. Study rooms along with projectors and whiteboards have all been provided by KEA.

#### Hostage/ResourceStakeholder: Kristian, App Academy

App Academy as a Stakeholder:

Kristian fits multiple roles he’s both a hostage and a resource stakeholder simultaneously. We’ve also argued that during the inception of our project, he was a Grey Eminence, dictating the terms and requirements for the project. He no longer fits that role due to the stage the project’s at. He is a hostage in term’s of him not being an active participant in the elaboration or construction phase. Yet as a client, he will be immensely important during our transition phase, and thus he’s simultaneously a Resource Stakeholder.

AppAcademy’s attitude towards the project:

AppAcademy has a low activity in regards to the project, to them it’s a nice to have program. Not a core business requirement. Their motive is facilitation of administrating part of their core business. They are contributing with running reviews and certain design aspects.

AppAcademy’s potential for conflict:

AppAcademy has a high potential for conflict with KEA as they have to adjust their expectancies to the boundaries set by KEA.

AppAcademy’s resources:

They are not providing any resources, as the software will be running as a standalone client.

#### Hostage: Course Responsibles’

Course Responsibles’ as Stakeholders:

These are the employees at App Academy that hold courses. They have no influence on any parts of the project and will have to accept the final product the way it is.

Course Responsibles’ attitude towards the project:

Their attitude could be negative as the project will impact their usual workflow. The project risks getting derailed due to their unwillingness to use the software.

Course Responsibles’ potential for conflict:

There is a chance for conflict depending on how AppAcademy introduces it to their employees. The conflict not be with the Project Group though.

Course Responsibles’ resources:

They have the potential to provide precious feedback, which could be useful.

#### Resource Stakeholder: Project Group

Project Group as a Stakeholder:

We’re present in all 4 UP Phases: Inception, Elaboration, Construction, Transition. We’re within certain limits defined by our Grey Eminence, in charge of the entirety of the project as the success or failure of it, falls upon us.

Project Group’s attitude towards the project:

Our attitude is that the project has ultimate importance. We will be the primary driving force in seeing that the project becomes realized and deployed. Our motive for this is both in learning and educational progress and we will thus be dedicating all of our time towards this.

Project Group’s potential for conflict:

There is a great risk for potential conflict with most of the other Stakeholders. KEA will hold great power over our project and as such, these kind of relations can become quite strenuous if not tackled in a constructive way. It is thus imperative that the Project Group maintains a positive attitude towards KEA and external influence in general.

Project Group’s resources:

We will be pulling from our knowledge in Software Development in order to create the project. We will be contributing most of our time and personal computers to see it realised.

#### External Stakeholders: Testers

Testers as Stakeholders:

These are our class mates, that have agreed to help us with testing the product. At first glance they don’t hold any influence on our project, but depending on feedback from them, they could become very important all of a sudden, and their influence upon our direction for the project could dramatically increase.

Testers’ attitude towards project:

Generally positive and their main contribution will come from their feedback as fellow students testing our software.

Testers’ potential for conflict:

Competitiveness could be an issue, but at this stage there has been no sign thereof. Again it will be important that the Project Group remains positive even if harsh feedback is given.

Testers’ resources:

Per se, the Testers will be contributing with their time as a Resource.

### Feasibility Studies

#### Technical Feasibility Study*[[1]](#footnote-1)[[2]](#footnote-2)*

* Hardware
  + CPUS: Intel 64-bit x86 Arch.
  + Computers: 4 Student Laptops
* Software
  + Java: JDK, JRE
  + Operation Systems: Windows 7-8-10, Linux Debian Based, Mac OSX
  + Database: MySQL
  + IDEs: JetBrains IntelliJ – DataGrip, Workbench
* Manpower
  + Developers: Emil, Jonas, Dennis, Fredrik
  + Testers: Class Members
  + Reviewers: Kristian (Client)
  + GUI Specialist: Dennis
  + Extra Curricular Skills: C, SML, Python, Ruby, PHP, HTML 5

Risk Assessment

* Failure to attain expected benefits from the project: Not a Risk due to the circumstance that we are Students and any learning process is beneficial, to an extent.
* Cost Estimates: We will have no costs to execute this project, material needed is available at no charge from our school.
* System Performance Levels: There are no specific performance requirements, but of course we don’t want an unstable system.
* Integration: There will be no software/hardware risk since the software will be system independent[[3]](#footnote-3).

#### Team Thoughts concerning Technical Feasibility

We have taken into consideration hardware and software requirements and have come to the conclusion that there won’t be any issues regarding these. In terms of abilities, we feel we are adequately equipped to implement all features for this project. However we have a small concern regarding our planned implementation of the “Course Certificate Generator” in terms of feasibility, but feel that this is a minor issue and no hinderance for us to proceed with the project. We have other solutions ready if needed.

#### Schedule Feasibility Study*[[4]](#footnote-4)[[5]](#footnote-5)*

We’re going to Budget our Schedule by breaking down the hours available to us.

Dates where we will be working on the project:

* April: 19, 22, 26, 29 / May: 3, 6, 10, 13, 17, 20, 24, 27, 31[[6]](#footnote-6)

We count 10 hours per date, which amounts to 130 hours per Developer and 520 man hours total.

These dates are extra curricular and don’t take into account the hours that will be gathered from class schedule.

There is a remaining 6.5 Weeks worth of class that will be spend on this project. 1 week worth of class hours is set to 17 hours. This amounts to 110.5 per Developer and totals to 442 man hours. This gives us a budget of 962 man hours and just 240 hours per Developer. This budget doesn’t take into account sick leave nor breaks.

We’ve set sick days a bit lower than national private sector average of 6.4[[7]](#footnote-7) to 5, because the project is scheduled outside flu season. The normal working hours for an employee is 37 hours per week, 7.4 hours per day. There is a total of 5 weeks vacation as a minimum which equates to 47 working weeks per year. Total amount of working hours per year is 1739, which translates to 235 days.

We now deduct 5 sick days from 235 total working days, which equates to 2.12%. Now apply 2.12% to our total developer hours of 240.5 which means we roughly get 1 sick day per developer during the course of the project, this results in a budget reduction of 4, total man hours is down to 958.

As a remainder we will also have to take breaks into account. We’ve set our preliminary limit at 1 per 4 hours of development time. 20% of off our budget of 958 equals 766. This means that we have a total of 191.5 hours dedicated to break time. Our budget is now reduced to 766 man hours, effectively 191.5 per developer, this number is real estimated work hours.

Budgeted hours:

|  |  |
| --- | --- |
| Segment | Developer Hours: 191.5 |
| ITO | 20 |
| SWD | 70 |
| SWK | 30 |
| OSCA | 10 |
| Budget | 61.5 |

We have decided to distribute 130 hours on our 4 primary project groups ITO, SWD, SWK and OSCA. This has left us with 61.5 hours as a buffer that can be applied where necessary. We think this is good since we’ve thus taken roughly 35% of our total developer hours and dedicated those to unforeseen events. We don’t think it would’ve been a prudent choice to plan with a 100% of available hours.

#### Team Thoughts regarding Schedule Feasibility

We decided to take a rather stringent approach to our schedule, seeing that making our total resources visible to us through numbers and facts would make it easier for us grasp and plan ahead.

Currency, finances, those were both reoccurring thoughts here, we decided to translate that into man hours, since this is the only resource that we will have to take into account. Another thing to remember here is that, we’ve decided to do the entirety of this project congruently as 4. Thus our focus is on our total developer hours of 191.5 and not the 766 man hours. This could be rather misleading if one was not aware. This decision means that we will limit our primary resource: time. We feel though as the development of this project is both a feat in productivity it is also a learning progress. By combining our abilities and knowledge we will gain a lot through shared knowledge and learning speed. Last but no least; faster productivity by combining efforts towards same goals constantly. Not one project member will be in the dark about any part of the project. Schedule wise, the project seems feasible at this stage, even with the consolidation of our man hours.

### Risk Assessment**[[8]](#footnote-8)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Risk | Probability | Consequence | Product | Preventive Measures | Responsible | Solutions | Responsible |
| Illness | 2 | 1 | 2 | There is not much we can do here | Project Dev’s | The rest of the team will have to compensate for the missing know-how. | The group |
| Lack of testing | 1 | 7 | 7 | Continously test so that it’s constantly maintained | The group | Implementing a rule that every method will have to be tested as it’s coded. | The group |
| Implementaiton | 1 | 10 | 10 | Maintain a dialogue with Kristian | Kristian | Make sure we document our code and user manual correctly. | The group |
| Resources Missing | 4 | 3 | 12 | Plan ahead. | The group | Have more than 1 plan to a problem | The group |
| Bad Estimates | 5 | 7 | 35 | Maintain our Estimates after each meeting/session. | The group | Adapt to changing circumstances, be dynamic. | The group |

#### Risk Analysis Comment[[9]](#footnote-9)

**Illness**: Due to the nature of how we’re working on this project, there is a very minimal chance of Illness impacted since we’ve pooled our resources, it basically means that we would all have to fall sick simultaneously for illness to severely impact us. It is highly unlikely that this will happen.

**Lack of testing**: It is very important that we continuously test our software, so that we’re sure that everything works as intended. To minimize risk here, we will be looking to unit-testing as a tool.

**Implementation**: It will be hard to do much during implementation other than introduce Kristian to our product in the best way imaginable. Since he can code, this means that we should comment and document our code adequately so that he can trouble-shoot himself. Secondly we will have to make sure our user-manual is well done, so that “Project Responsibles” can properly solve their issues by consulting it.

**Resources Missing**: In case of computers crashing as an example it’s important that we have backups of our code. We’ve decided to use GitHub to manage the versions of our software and for live backup we’re using Dropbox. This way our code is located on several different systems and maintained there, preventing irrecoverable system crashes.

**Bad Estimates**: Considering that we’re still students with a limited knowledge and experience, it’s hard for us to “guess-timate” in a correct manner. Therefor it’s imperative that we meticulously adjust and maintain our budgeted time schedule. If things need to be speeded up we have to be prepared to do so in order to deploy a good product.

## Elaboration

### FURPS+

#### Functional

##### Features

The program will be a standalone application that have the features to create certificates, certificate templates and courses attaching material, add course responsible and participants, send the certificates using an SMTP server. The material will be uploaded through FTP.

Generation of certificate templates will use “Drag ‘n’ Drop” functionality to place placeholders.

##### Security

The security have low priority since it mainly lies on server side and since it’s only trusted users that can do any real damage to the system. The database is backed up so if it is needed it is recoverable.

Regarding exploits, the database queries will be sanitised and only current administrators can create other responsibles, and administrators, giving no privilege escalation.

#### Usability

##### Human Factors

No flashing colors or rapidly changing color schemes due to the users might have photosensitive epilepsy. The user interface aims to be consistent and distinguish between database operations and the program's own interface.

##### Documentation

Documentation consists of code commenting when the source code is supplied and the use of the program itself is documented with a manual.

A javadoc is also supplied with the source code.

##### Help

Error messages should be displayed in common English and not messages like “error: 45”. If incorrect information is entered the user should be made aware of it (such as “abc” where a number is required”

#### Reliability

##### Frequency of failure

We expect the rate of failures will be low due to high error handling and few places that can go wrong. The failures that might take place would be loss of internet connection, missing settings file or permissions.

##### Recoverability

If the settings file is missing then it will be created with default values which then can be altered. If the user doesn’t have the appropriate file permissions the user need to contact their computer administrator to get these right, which should be in the user’s own home.

If the internet connection is lost, then a reconnection will solve the problem.

Corruption in the database will be recovered by backing up the database frequently so not all will be lost, but only changes since last backup.

##### Predictability

Usually it isn’t predictable when the user might or might not have access to the internet.

Altering the file permissions or deleting the settings file is extremely unpredictable and normally never happens.

#### Performance

##### Response time

If the program has to wait for a response from an external source, then there is a timeout when the program has waited for X seconds. Showing new views is instant (as fast as the computer can handle), however if data from an external source should be displayed in the view, then the program waits for the data before displaying the new view. A message or “loading screen” is displayed in the meantime.

##### Resource usage

The programs resource usage is of a basic program using javafx running in java virtual machine. The program will not perform any actions if the user is idle.

It will be scalable because there are no computations. For creating courses, participants and responsible will be independent of how many of them there are in the database.

Viewing the rows will be dependent on the amount of rows but is optimized in the database.

#### Supportability

Swappable language file is integrated if there is enough time to do that.

#### Legal(+)

Neither the participants' nor the course responsibles' CPR number is stored, which means that Danish law about storing data that can identify the physical person.

The module for using FTP is licensed under the Apache License.

### Use Cases

#### Fully Dressed - Generate Certificate Template

**Primary Actor:** Course Responsible w/ Admin Rights

**Stakeholders and Interests:**

**-Course Responsible:** Wants to effectively generate a usable certificate template.

**-Company:** Course certificates must be retrievable at any point. Scalable amount of templates must be storable.

**Preconditions:**

a. Course is logged in and authenticated.

b. Systems are operational (Database and our software itself).

c. Course Responsible has a bmp/png/jpeg picture ready to upload into the system

d. Course Responsible knows where the Settings tab is and is familiar with the system in general.

**Post conditions:**

a. Template is stored and added persistently.

**Main Success Scenario:**

1. Course Responsible logs into system.

2. Course Responsible clicks into ‘Settings’ pane.

3. Course Responsible clicks ‘Add Course Certificate Template’.

4. ‘Generate Course Certificate’-Windows pops-up.

5. Course Responsible selects necessary certificate information from:

- ‘Signature of Course Responsible’

- ‘Date of Certificate Print’

- ‘Course Participant’

- ‘Course Name’

6. Course Responsible proceeds to upload certificate IMG via ‘Insert Certificate IMG’.

7. Course Responsible positions selected certificate information correctly on Certificate IMG

8. Course Responsible Inputs Certificate Template Name via ‘Insert Certificate Name’.

9. Course Responsible loads correct font for template. Default ‘Times New Roman’.

10. Course Responsible verifies that everything is correct and proceeds to click ‘Confirm’ button.

11. Course Responsible is thrown back to ‘Settings Pane’ and template is now usable.

**Alternate Flows:**

1. Course Responsible fails to login.

1a. Course responsible retries with correct information.

1b. Login successful, course responsible is processed to Home Screen.

1c. Course Responsible fails, all accounts have to be reset.

1. Course Responsible fails to login.

a. Course Responsible retries with correct information.

b. Login successful, admin is processed to Home Screen.

1. Login fails, admin has to reset all admin accounts.

2. System crashes

a. Admin is forced to restart the system.

4. Pop-Up window doesn’t come to front; admin is convinced that the window hasn’t popped-up.

a. Admin restarts system, assuming failure.

b. Admin realizes error and manually retrieves windows.

1. Admin proceeds with step 5. in Main Success Scenario.

5. Admin refuses to input necessary information.

a. Admin closes window.

b. Admin concedes and inputs mandatory information.

6. System fails to recognize file.

a. Admin tries again; system acknowledges IMG file.

b. Admin fails to upload an accepted format.

c. Has to exit system until error has been corrected.

8. Admin uses an unknown ‘char’ in input field.

a. System refuses input.

b. Admin corrects input and resumes.

9. Admin uses a font type/file, that isn’t recognized by java library.

a. He is prompted to input a correctly formatted file.

11. System crashes during safe.

a. Admin verifies whether data was saved anyway.

b. Data wasn’t saved and Admin has to retry.

**Special Requirements:**

-Admin has an OS that’s supported JVM 8(Java Virtual Machine).

-Admin has IMG file for certificate. See steps 6 and 6ab.

-Admin has peripherals: Keyboard, Mouse, Monitor.

-Admin has access to database file.

-Admin has access to his OS file system.

**Frequency of Occurrence:** As per meeting with Client, it would be infrequent.

**Open Issues:**

-File type variations.

-Font type variations.

-OS GUI variations (example. Arch Linux).

#### Fully Dressed: Send Missing Certificate

**Primary Actor:** Course Responsible

**Stakeholders and interests:**

**Course Responsible:** Wants to send certificates to the course participants who haven’t received them yet.

**Course Participant(s):** Wants to receive the correct certificate from their attended courses.

**App Academy:** Wants the other stakeholders’ goals to be fulfilled properly.

**Client Company:** Wants their employees to get the proper certification.

**Preconditions:**

A course has been held and certificates have to be sent.

The course responsible who uses the system does not accidently close the program.

The course responsible does not send certificates to the wrong course participants.

**Post conditions:**

Course certificates have been sent to the correct recipients. Certificate is stored on ftp-server.

**Main Success Scenario:**

1. Course Responsible logs into system.

2. Course Responsible selects "Courses" pane.

3. Course Responsible arrives at table view of 'Courses'.

4. Course Responsible presses 'View’ at the Course he wants to send certificates for.

5. Course Responsible arrives at a table view of 'Course Participants'.

6. Course Responsible presses ‘Missing Cert.’ and is presented with a filtered List of Course Participants that haven’t received their certificate.

7. Course Responsible presses ‘Send’ for desired Course Participants.

8. Course Responsible closes the table when done.

**Alternate Flows:**

1. Course responsible cannot log in.

a. Course responsible may have entered the wrong login information and therefore retries with the correct username and password.

2. The course responsible cannot find the “Courses” pane.

a. The course responsible filters through the panes until he finds the “Courses” pane

3. Course responsible cannot click Courses pane.

a. The course responsible attempts a system restart.

b. Course responsible can now use the pane as normal.

4. Course responsible does not arrive at the “Courses” table view.

a. Course responsible ensures he/she chose the right pane. If this isn’t the case, he/she can use the previous window button on the option page he/she landed on.

b. Course responsible can now choose the right path to the right pane

5. The course responsible cannot find the course where certificates have to be sent.

a. Course responsible looks through the columns until he/she finds where the “View” buttons are located.

b. Course responsible can now click the view button.

6. Course responsible does not arrive at the “Course participants” table view.

a. Course responsible ensures he/she chose the right course. If this isn’t the case, he/she can use the previous window button in the table view he/she landed on.

b. Course responsible can now choose to view the right course and thereby arrive at the table view.

9. Course responsible cannot close the table view.

a. Course responsible may try to force quit the program, not using the inbuilt quitting feature.

b. The course responsible has now successfully quit the program.

**Special Requirements:**

-Course Responsible has an OS that’s supported JVM 8(Java Virtual Machine.

-Course Responsible has peripherals: Keyboard, Mouse, Monitor.

-Course Responsible has access to internet/intranet.

-Course Responsible has access to his OS file system.

**Frequency of Occurrence:** Frequent, every time a course has been held, certificates will have to be send.

#### Fully Dressed: Add Course

**Primary Actor:** Course Responsible

**Stakeholders and Interests:**

**-Course Responsible:** Wants to create a course without errors and a minimal amount of selection options.

**-AppAcademy:** Wants material used for each course to be stored for future use and traceability.

**Preconditions:**

a. Course Responsible is logged in and authenticated.

b. Systems are operational.

c. Course Responsible Has Course Material ready in a zip file and is aware of its location within his file system.

d. Course Responsible knows where the Courses tab is and is familiar with the system in general.

**Post conditions:** Course is saved correctly in database.

**Main Success Scenario:**

1. Course Responsible arrives at Home Screen.

2. Course Responsible selects ‘Courses’ pane.

3. Course Responsible clicks ‘Add New’ button.

4. Course Responsible fills out the form (Name, Course Start Date, Course End Date, sets Course Responsible, Uploads Course Material)

5. Course Responsible clicks ‘Add Course’ button.

6. Course Responsible is now returned to ‘Courses’ Pane.

**Alternate Flows:**

1. System crashes after login.

a. Course responsible is forced to restart the system

2. Course responsible cannot find the “Courses” pane.

a. Course Responsible manually looks through the panes until he/she finds the “Course” pane and then selects it.

3. The “Add new” button does not work.

a. Course responsible goes back to the previous window or force restarts the system. Either way she ends up at the “Courses” pane once more.

b. The “Add new” button should now work as intended.

4. Course responsible fails to fill in the required fields needed for the course creation.

a. Course responsible finds the field(s) where he/she forgot to fill in the necessary information.

b. The course can now be created.

5. Course responsible cannot click the “Add Course” button.

a. The “Add Course” button cannot be interacted with as long as the necessary information needed to create a course hasn’t been added. Therefore, see step 4. a.

b. If the button still doesn’t work, see step 1. a.

c. The “Add Course” button is now functional.

6. The “Course List” view does not show up.

a. See step 1. a.

b. Verify the course has been added correctly.

**Special Requirements:**

-Course Responsible has an OS that’s supported JVM 8(Java Virtual Machine.

-Course Responsible has peripherals: Keyboard, Mouse, Monitor.

-Course Responsible has access to internet/intranet.

-Course Responsible has access to his OS file system.

**Frequency of Occurrence:** Frequent, this will happen every time AppAcademy has a course.

**Open Issues:**

What if there isn't any Course Material to upload?

#### Casual: Remove Course

**Primary Actor:** Course Responsible w/ Admin Rights

**Stakeholders and Interests:**

**-Course Responsible:** Wants to remove a Course as easy as possible.

**-AppAcademy:** Wants to maintain data integrity, Courses shouldn't be too easy to delete.

**Preconditions:**

Course Responsible has retrieved correct CourseID for the course he wants to remove.

**Main Success Scenario:**

1. Course Responsible successfully authenticates.

2. Course Responsible proceeds to 'Settings' pane.

3. Course Responsible clicks 'Remove Course' button.

4. Course Responsible is prompted with a Pop-up where he inputs CourseID and confirms his choice to remove Course by clicking 'Confirm' button.

5. Course Responsible is prompted with a confirmation message and has to press 'Okay' to proceed.

6. Course Responsible is returned to 'Home Screen'.

**Alternate Success Scenario:**

4. He inputs incorrect CourseID and is prompted with an error message.

a. He retries with correct CourseID.

#### Casual: Remove Course Participant

**Primary Actor:** Course Responsible

**Stakeholders and interests:**

**- Course Responsible:** Wants to remove a course participant without any trouble.

**- App Academy:** Wants their courses to consist of the assigned participants.

**Preconditions:**

Course Responsible has successfully authenticated and landed on home screen.

**Main Success Scenario:**

1. Course Responsible opens the “Course Participants” pane.

2. Course Responsible selects desired “Course Participant”.

3. Course Responsible clicks the “Remove” button.

**Alternate Success Scenario:**

1. The course responsible does not open the right pane

a. The course responsible returns to the home screen and tries again

2. The “Courses” pane does not work

a. The course responsible restarts the system.

#### Casual: Change FTP Settings

**Primary Actor:** Course Responsible w/ Admin Rights

**Stakeholders and Interests:**

**-Course Responsible:** Wants to correct information so that he can access correct Course Material and Certificate Templates.

**-AppAcademy:** Wants to make sure that Course Responsible has access.

**Main Success Scenario:**

1. Course Responsible successfully logs in.

2. Course Responsible selects 'Settings' pane.

3. Course Responsible clicks 'Change FTP' button.

4. Course Responsible is processed to a new window with following fields:

a. Username

b. Password

c. Host

d. Port

He fills in correct information.

5. Course Responsible applies new information to the system.

6. He's processed back to the 'Settings' pane.

**Alternate Success Scenario:**

3. Course responsible doesn't have correct information.

a. He will contact App Academy for correct information.

#### Casual: Change SMTP Settings

**Primary Actor:** Course Responsible w/ Admin Rights

**Stakeholders and Interests:**

**Course responsible:** Wants SMTP settings to be correct so he/she can send out certificates.

**App Academy:** Wants to the SMTP settings to be correct so the Course Responsible can use the system.

**Main Success Scenario:**

1. Course Responsible selects 'Settings' pane.

2. Course Responsible clicks 'Change SMTP' button.

3. Course Responsible is processed to a new window with following fields:

a. Username

b. Password

c. Host

d. Port

e. Sender’s email

He fills in correct information.

4. Course Responsible applies new information to the system.

7. He's processed back to the 'Settings' pane.

**Alternate Success Scenario:**

4. Course responsible doesn't have correct information.

a. He will contact App Academy for correct information.

#### Casual: Change MySQL Settings

Primary Actor: Course Responsible w/ Admin Rights

Stakeholders and Interests:

Course Responsible: Wants MySQL settings to be correct so that aCerts is functional.

App Academy: Wants MySQL settings to be correct so that their Course Responsibles can work efficiently.

Main Success Scenario:

1. Course Responsible selects ‘Settings’ pane.
2. Course Responsible clicks ‘Change MySQL’ button.
3. Course Responsible is processed to a new window with following fields:
   1. Username
   2. Password
   3. Host
   4. Port
   5. Database Name
4. Course Responsible applies new information to the system.
5. He’s processed back to the ‘Settings’ pane.

Alternate Success Scenario:

4. Course Responsible doesn’t have correct information for the system.

a. Course Responsible contacts AppAcademy for correct information.

#### Brief: Add Course Participant(s)

Course Responsible logs into the system and arrives at home screen. Course Responsible clicks ‘Course Participants’ pane. He then selects ‘Add New’ and is prompted with a Pop-Up window, where he inputs prompted information fields. When finished he finalizes by pressing ‘Add Course Participant’.

#### Brief: Add Course responsible

Course Responsible with Admin Rights logs into the system. He proceeds to the ‘Course Responsibles’ pane and clicks ‘Add new’. From there he fills in mandatory data into the new Pop-Up window (First Name, Last Name, E-mail, Phone#). When information has been filled and a decision has been made whether or not this Course Responsible should have admin rights by checking the box next to ‘Admin Rights’; now he can press Add Course Responsible. The Course Responsible has now been added and he is moved back to Course Responsible table view where the recently added ‘Course Responsible’ should be present.

#### Brief: Remove Course Responsible

Course Responsible with Admin Rights logs into the system. He proceeds to Course Responsible pane. He is now in a table view with all Course Responsible. He selects the Course Responsible he wants to remove from the system and presses the Remove button found in the bottom left corner.

#### Brief: Remove Certificate Template

Course Responsible with Admin Rights logs into the system. He proceeds to Settings pane. Here he is presented with a dropdown from which he will select the desired template to be removed. He can now press Remove Certificate Template button and the Template will no longer be available as a selection when creating a new Course.

#### Brief: View Courses

Course Responsible logs into the system and arrives at home screen. Course Responsible clicks the “Courses” pane which leads to a table view of the courses.

#### Brief: View Course Responsibles

Course Responsible with Admin Rights logs into the system. He proceeds to the Course Responsibles pane; he is now presented with a view of all Course Responsibles that has active courses.

#### Brief: View Missing Certificate

Course Responsible logs into system and arrives at home screen. He now selects ‘Courses’ pane. Here he is presented with a set of filters on the left hand side of the table. He selects ‘Missing Certs’. The table is now filtered to only show courses that contains participants missing a certificate. Alternately: ‘Course Responsible’ clicks ‘Course Participants’ pane. Here he is presented with a table view of participants. On his left hand side there is a set of filters, from which he selects ‘Missing Cert.’ Now he is presented with a filtered list of participants missing their Certificates.

#### Brief: View Active Courses

Course Responsible logs into the system. From there he proceeds to the Courses pane. Here is presented with a table view of his courses. This table is set to present Active Courses automatically.

#### Brief: View all Courses

Course Responsible logs into the system. From there he proceeds to the Courses pane. Here is presented with a table view of his courses. This table is set to present Active Courses automatically so he has to press 'All Courses' to be presented with a view of all courses at AppAcademy.

#### Brief: View inactive Courses

Course Responsible logs into the system. From there he proceeds to the Courses pane. Here is presented with a table view of his courses. This table is set to present Active Courses automatically, so he has to select 'InActive Courses' filter in the upper left corner where the table will be filtered to show Courses that are from a previous date from todays' date.

#### View Course Participants

Course Responsible logs into the system. From there he proceeds to the Course Participants pane. He is now presented with a table view of Course Participants.

### System Sequence Diagrams



Figure . Generate Certificate Template

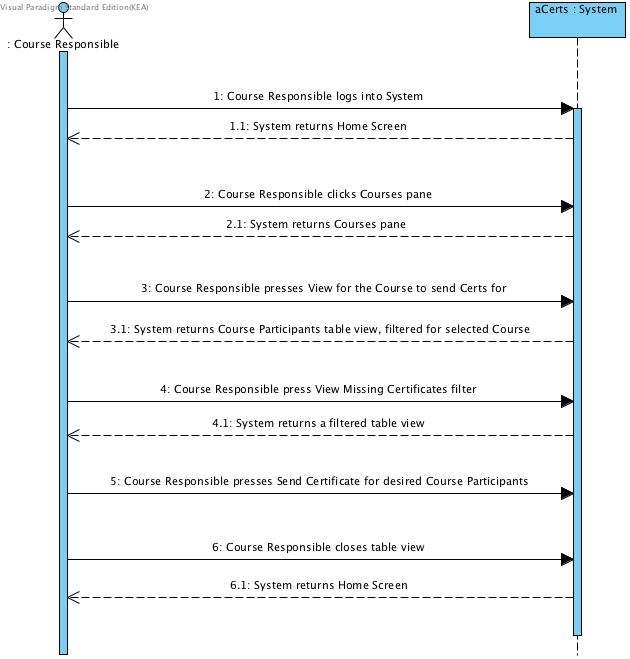


Figure . Send Missing Certificate

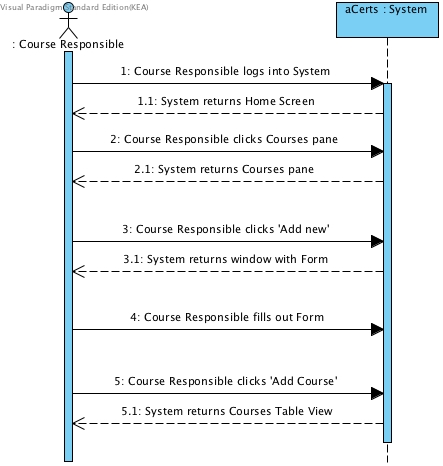


Figure . Add Course

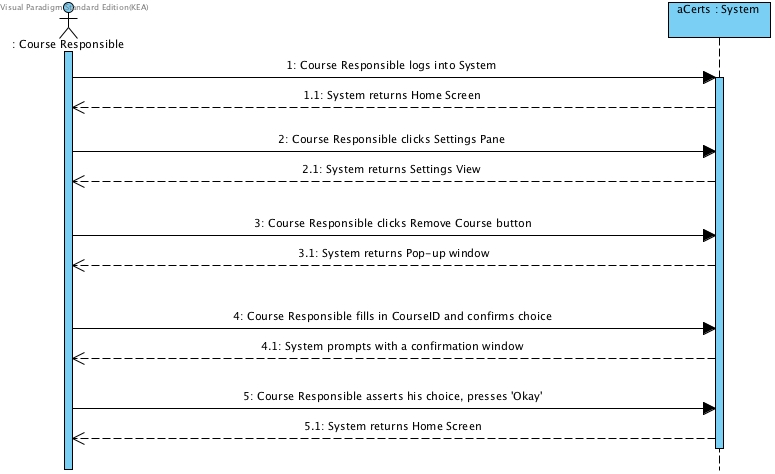


Figure . Remove Course

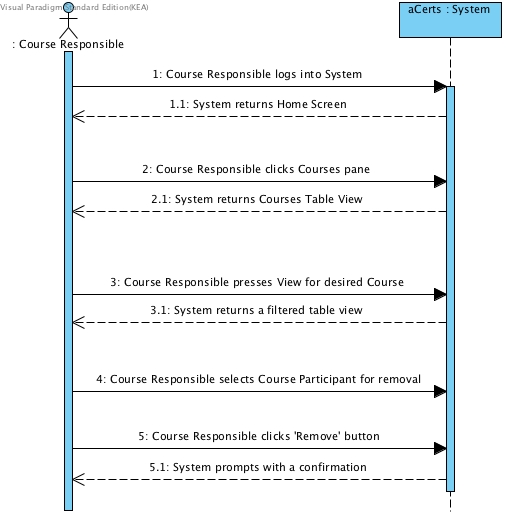


Figure Remove Course Participant

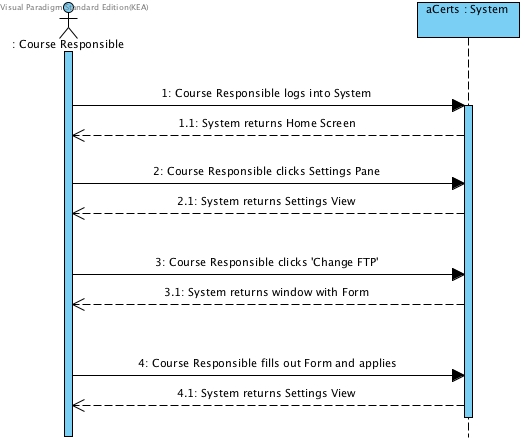
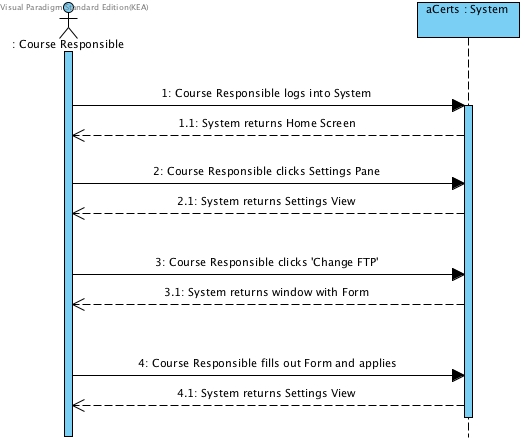


Figure . Change FTP Settings

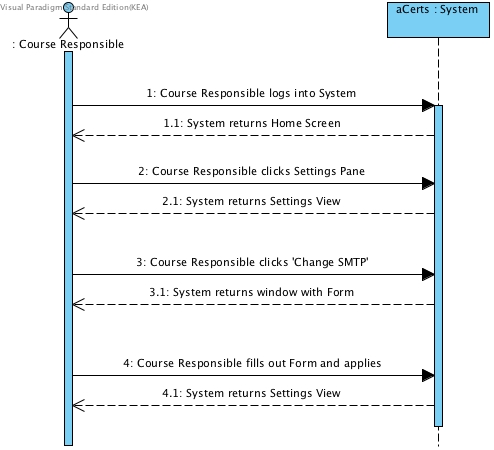


Figure . Change SMTP Settings

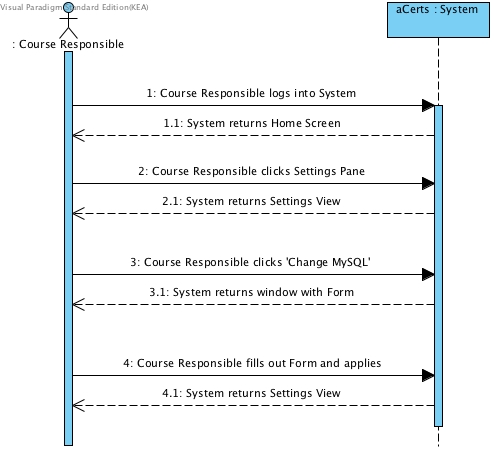
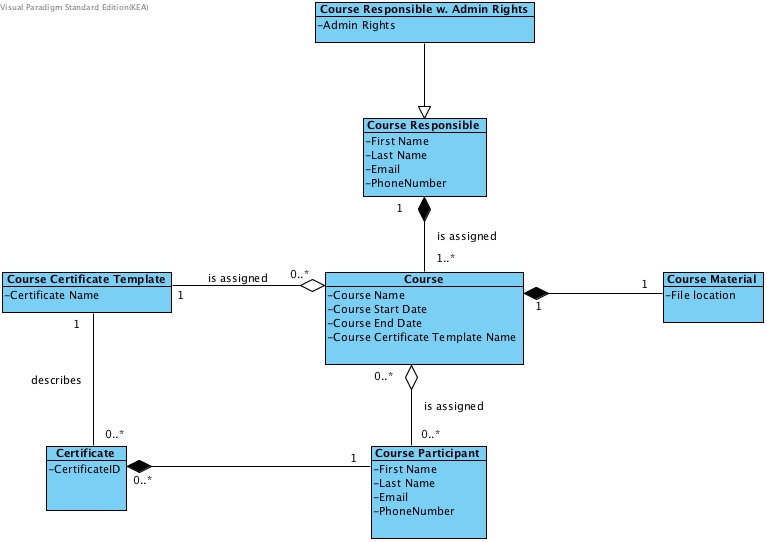
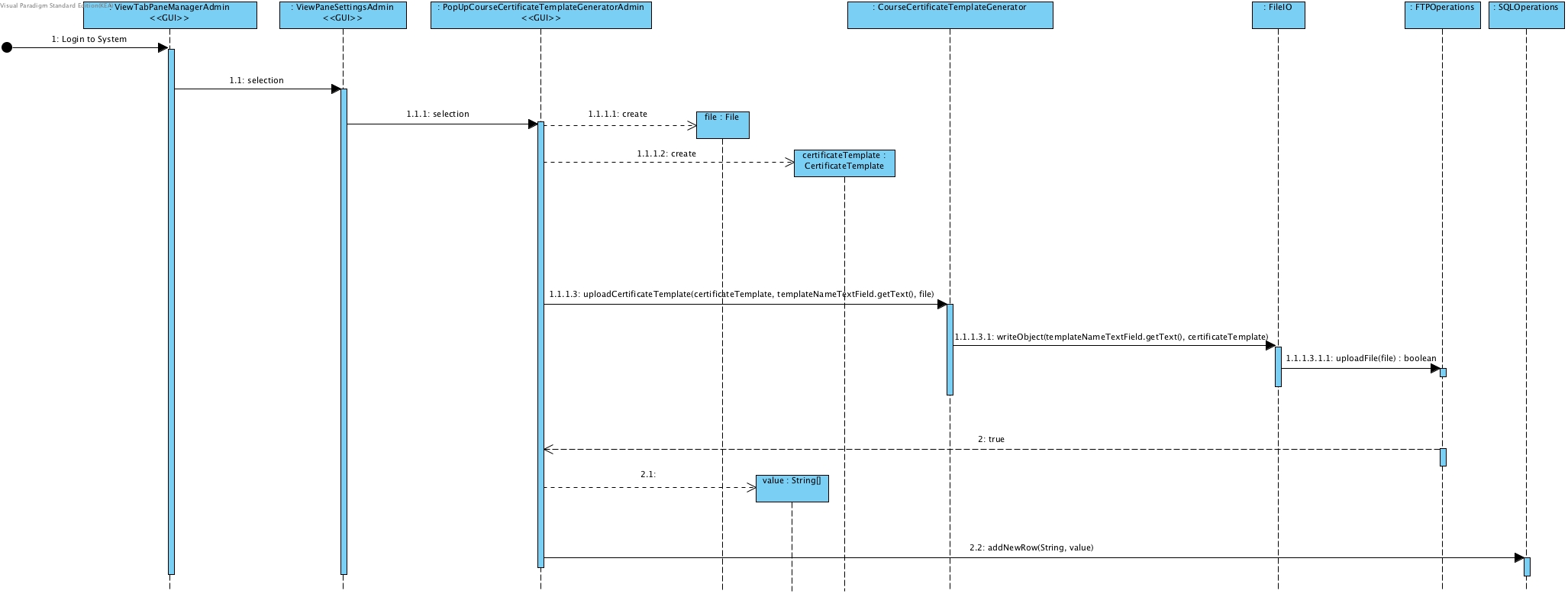


Figure . Change MySQL settings

### Use Case Diagram

Domain Model[[10]](#footnote-10)

### Sequence Diagram

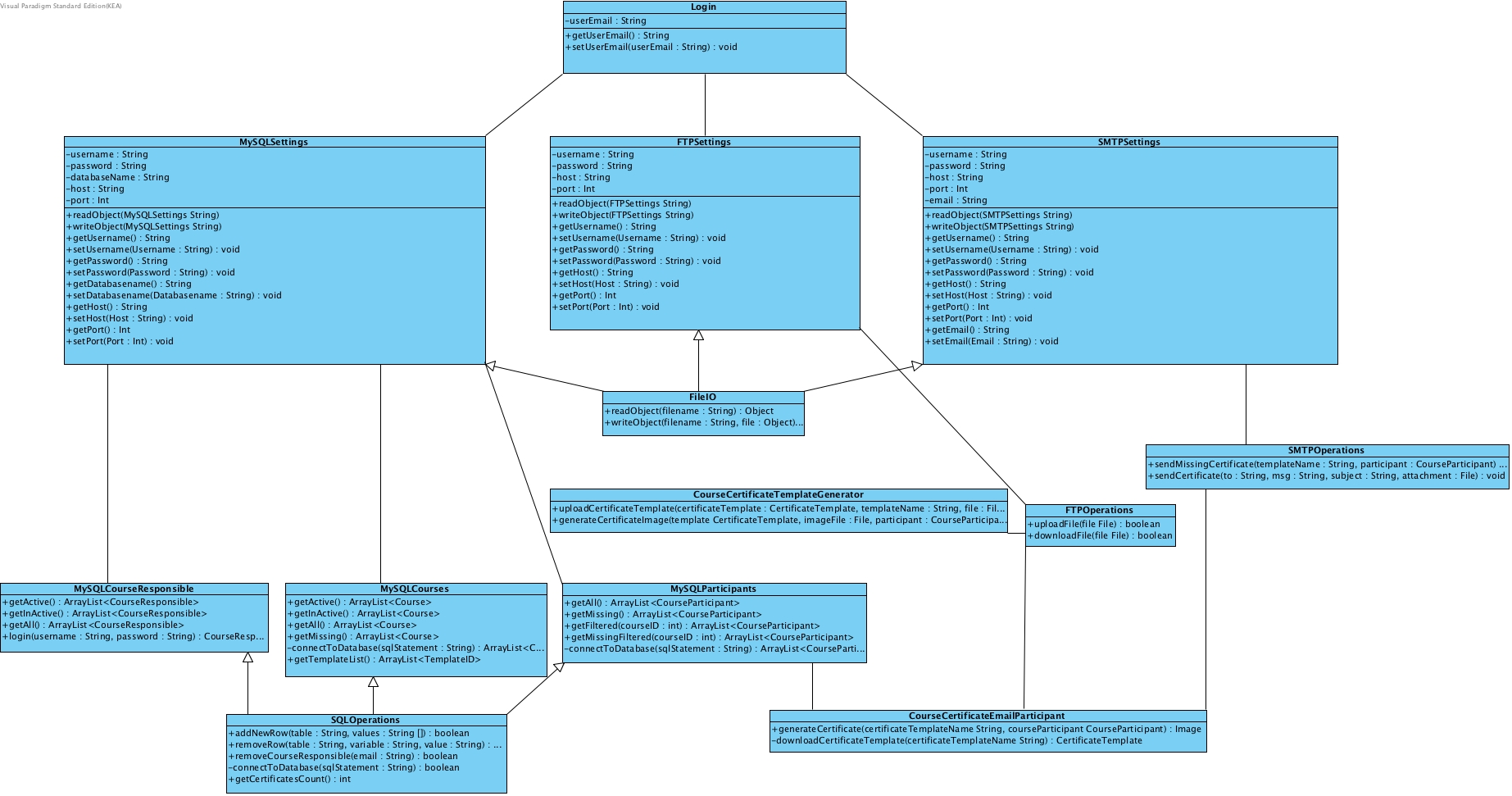


## Construction

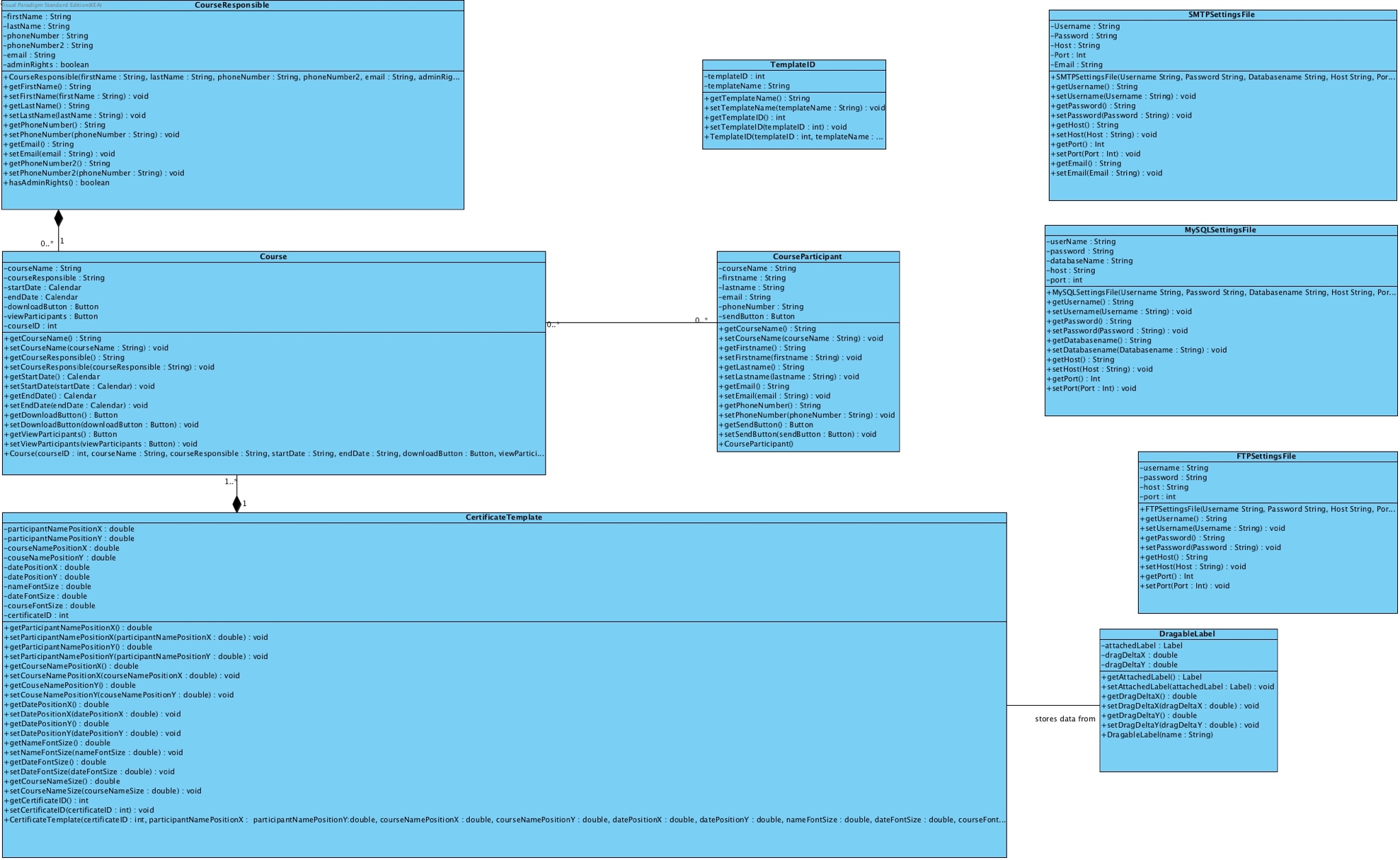
### Class Diagrams

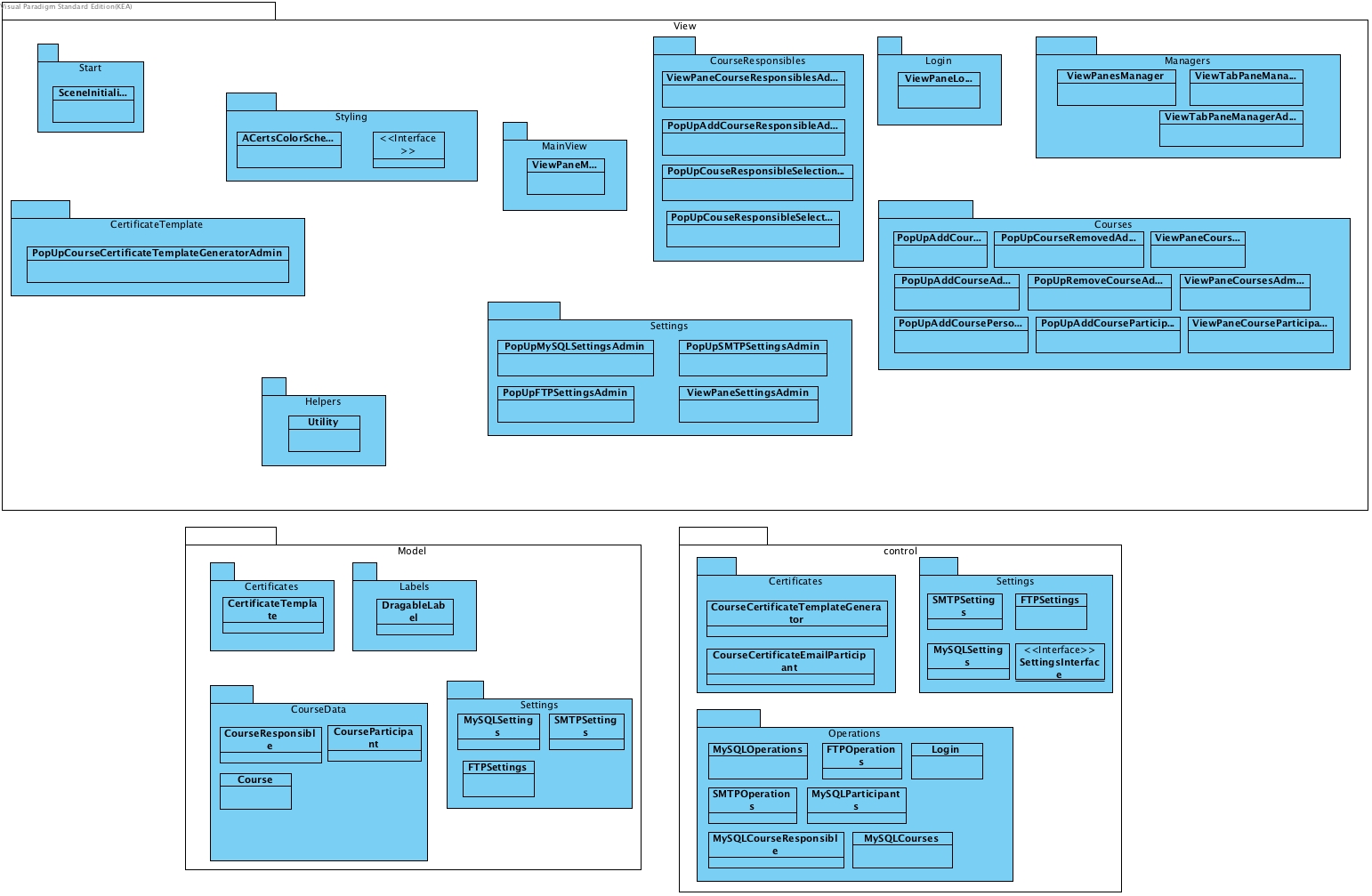
#### GUI Class Diagram

#### Functional Class Diagram



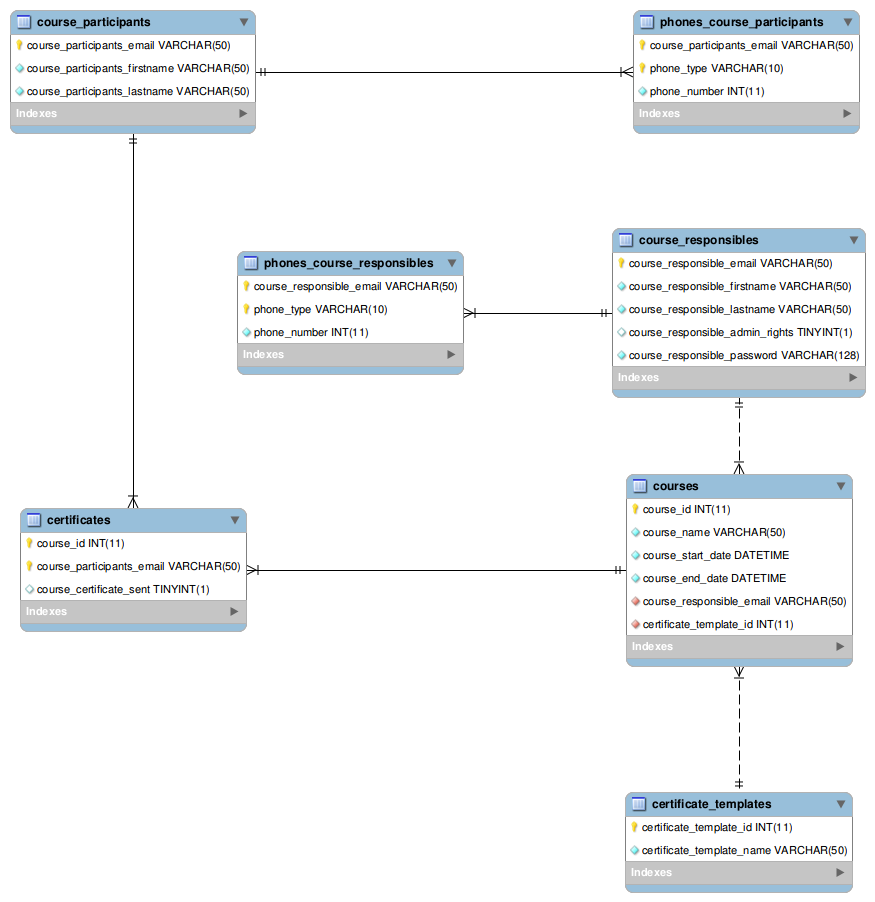
#### Model Class Diagram

Logical Architecture Diagram



### Database

### Entity-Relationship Diagram

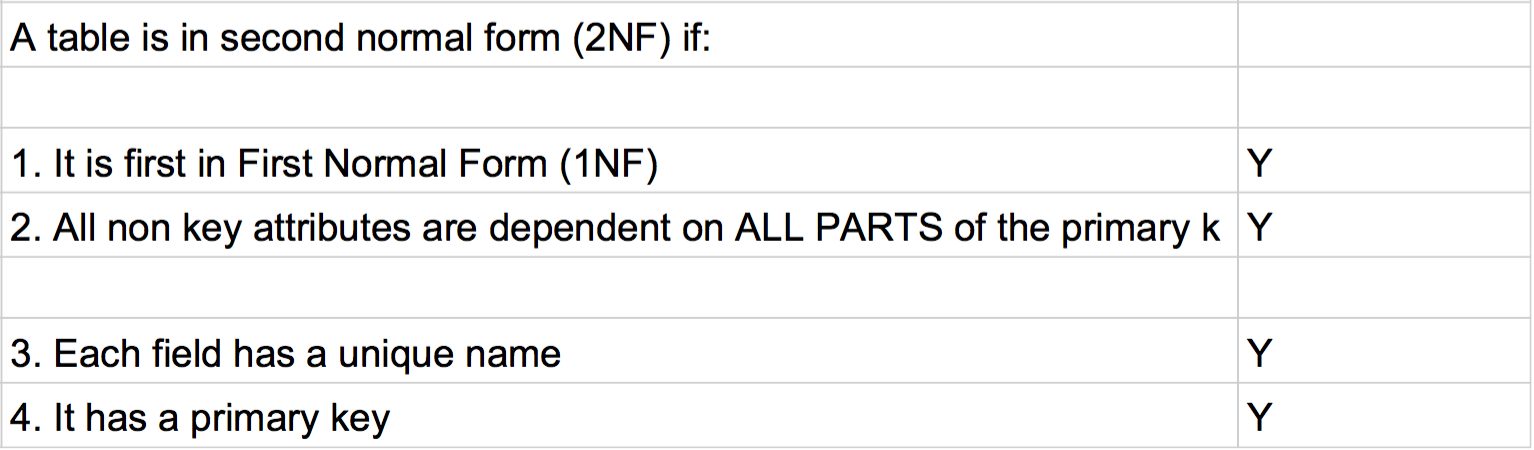


### Database Normalization Process

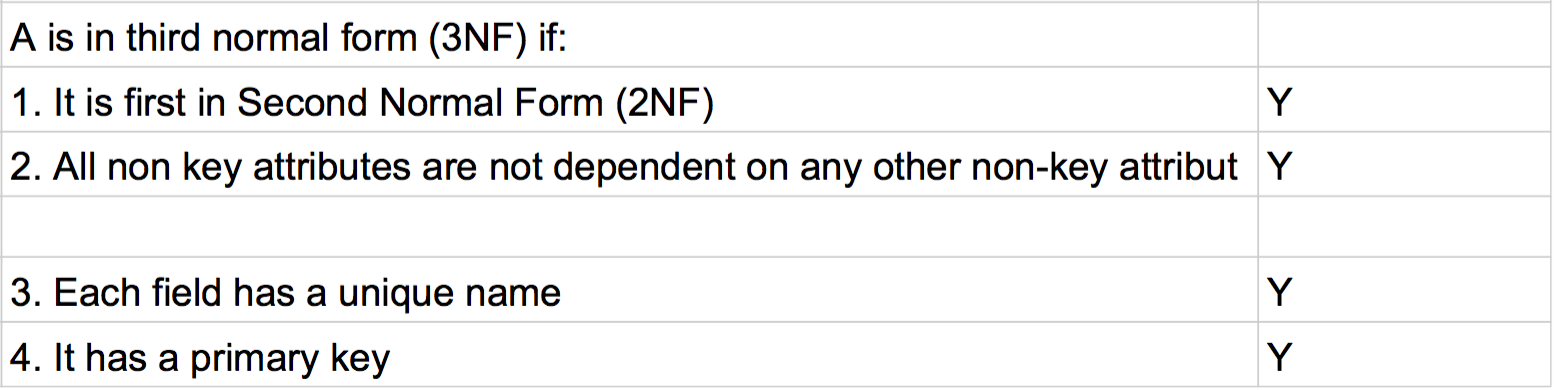
1st Normal Form – Verification Questions

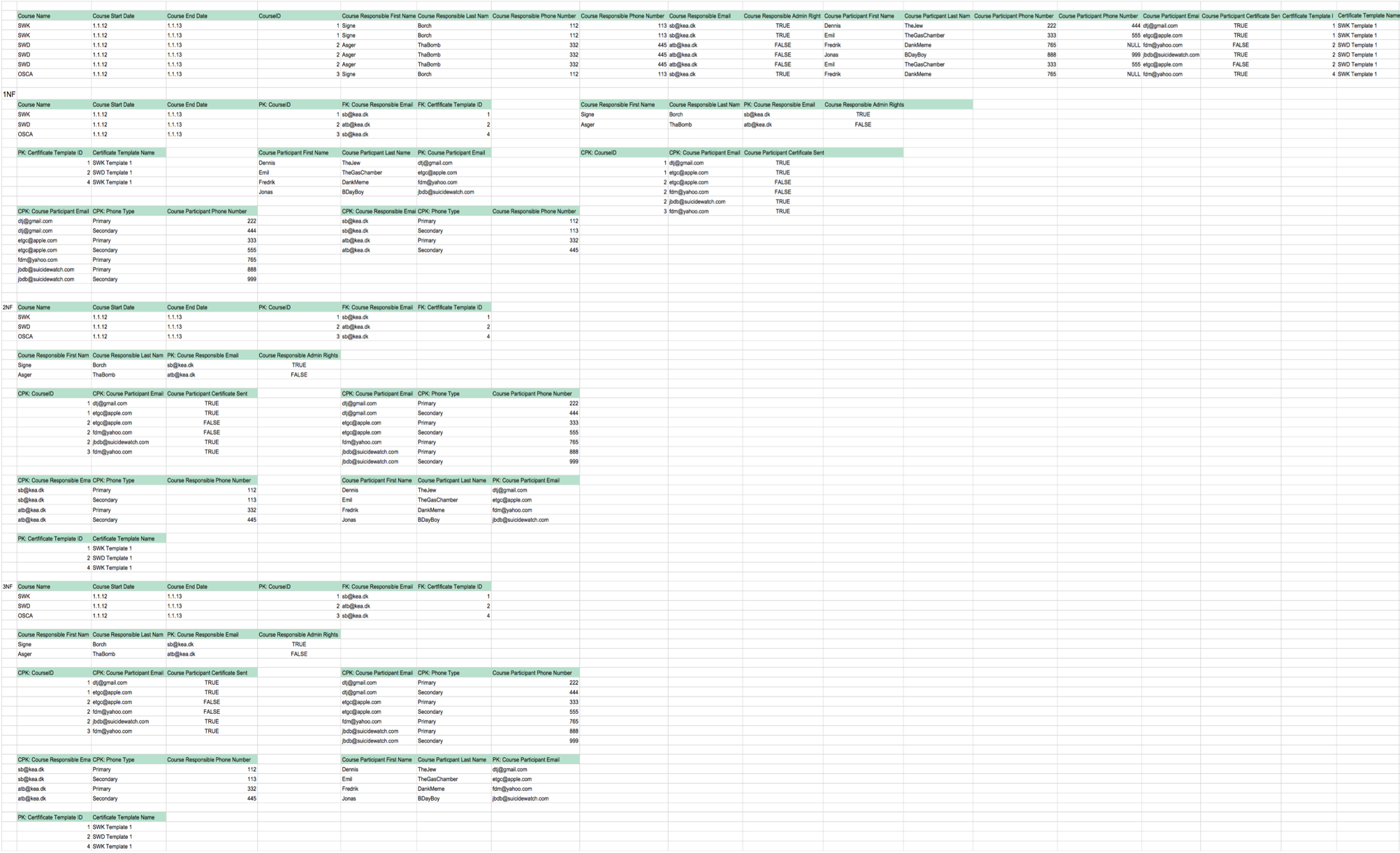


2nd Normal Form – Verification Questions



3rd Normal Form – Verification Questions







## Transition

### Reviews

#### Review by Kristian during 2nd Iteration

#### Review by class member Lukas

#### Review with Kristian for deployment

### Program User-Manual / AppAcademy Certificates Manager

#### First Time Setup

When running the program for the first time, the program will generate a MySQLSettings file.

Since there is not an existing MySQLSettings file then the program will connect to a

MySQL server at localhost, named 'AppAcademy' with the password 12345678.

Once the user has logged in these settings can be changed.

When logged in the first time, all FTPSettings and SMTPSettings are not set so alot

of features wont work until the user has set the FTP settings and SMTP settings.

If remote users should use the program, then all the settings files should be distributed

along with the program. It is important that the remote users do not get settings

which point to a localhost, if they are to use the same server.

#### Creating a certificate template

When creating a certificate template, you should have an image ready where is it possible

to write a participant name, date and a course name.

Afterwards you can edit the position of where the date, participant name and course name should be

written on the image. Drag the red markersaround with the example text to customise the certificate.

When positionning and sizing is done then give the template a name and click 'generate certificate' to store the template for later use.

#### Creating a new course responsible

Creating a course responsible is as simple as filling out the form that is presented.

Once the form is filled out, press the 'add course responsible' to create the course responsible.

If information is missing, then the field missing information is highlighted in red.

#### Creating a new course

When creating a new course you will need to have the course responsible that you want

to assign to the course in the system. You must also have a certificate template ready.

Fill out the form and then create the

course by pressing the 'create course' button. If it fails due to missing information, then

the required fields that are missing information is highlighted with red.

#### Creating a new course participant

When creating a new course participant you will have to have a course ready you can assign

the participant to.

The form which is presented should then be filled out. Afterwards pressing the 'add course participants'

will add the participant to the system.

#### Sending a certificate

To send a certificate you need to have a participant that is assigned a course. See the 'Creating a new course participant'

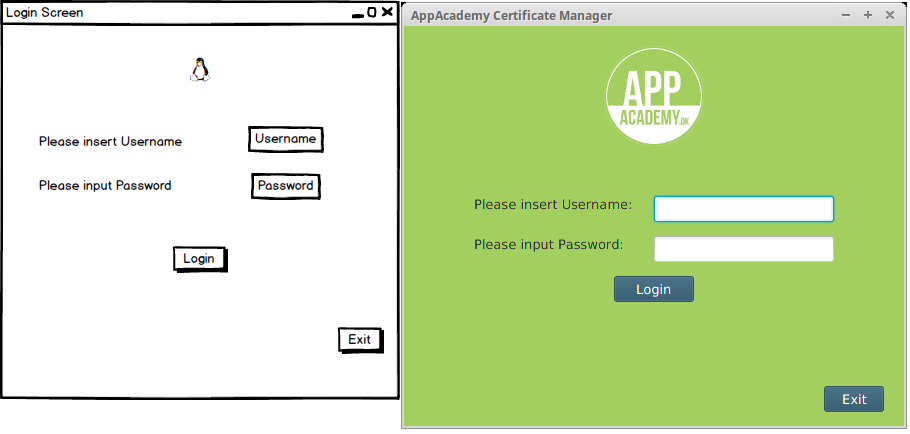
step on how to do so.

To send the certificate find the participant you want to give a certificate in the course partifipants tab,

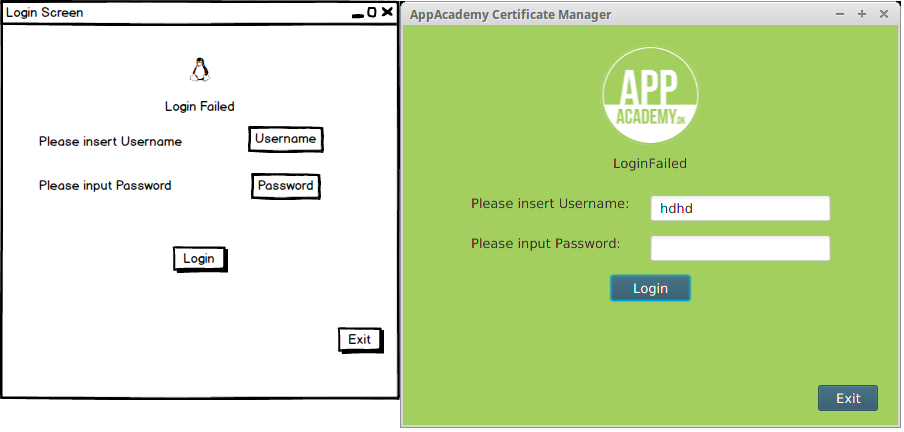
and press the send button.

## aCerts – Prototype contra actual Program

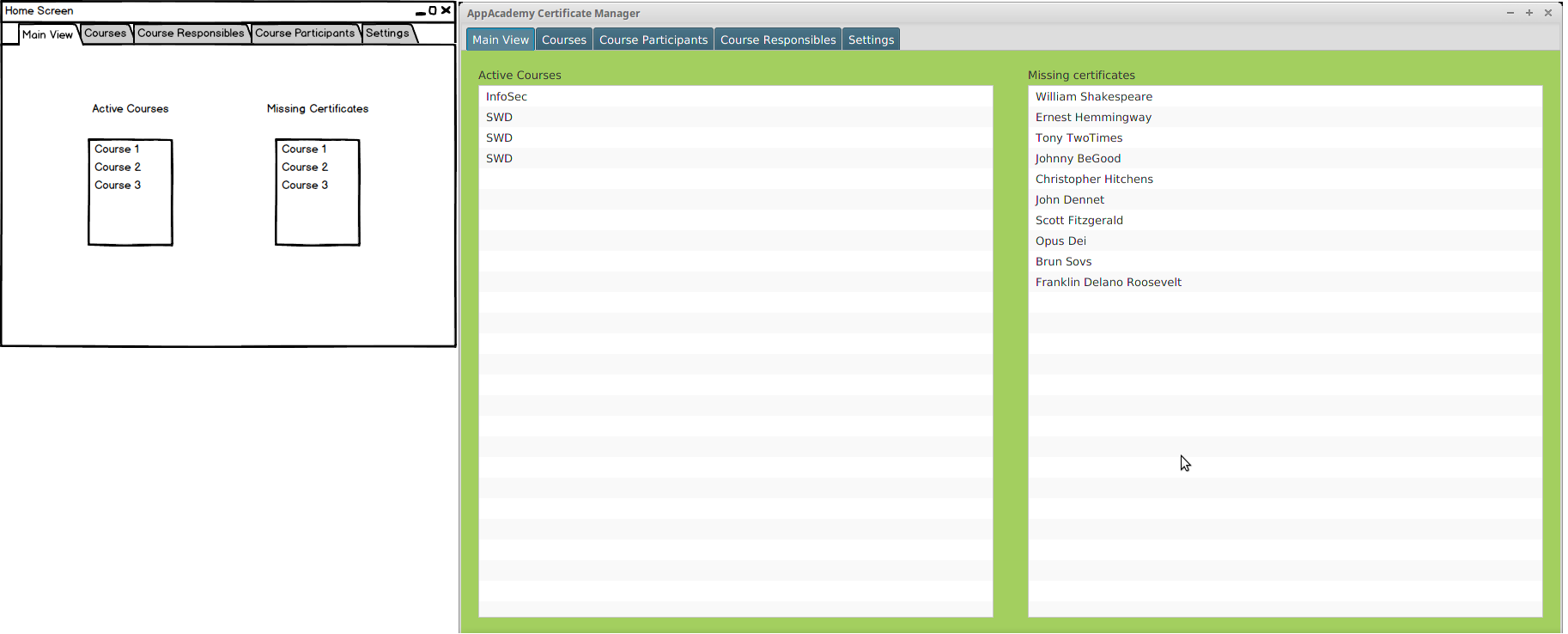
#### Login Screen

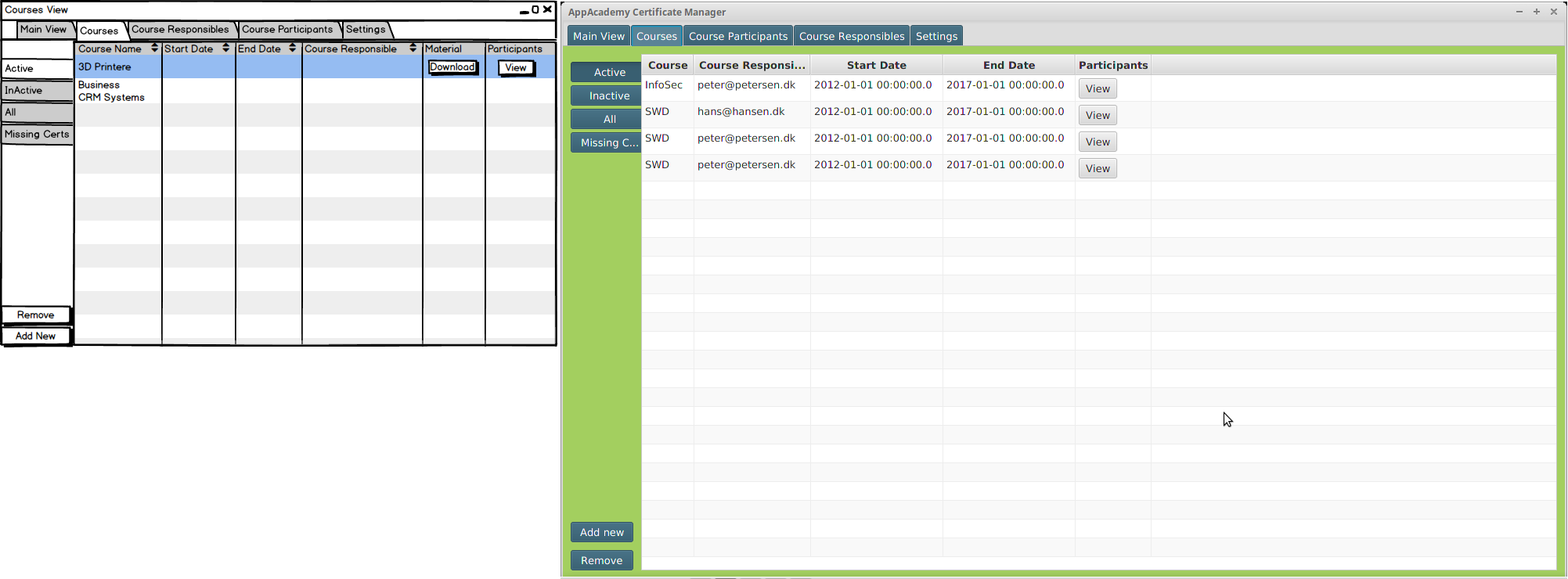


#### Login Failed

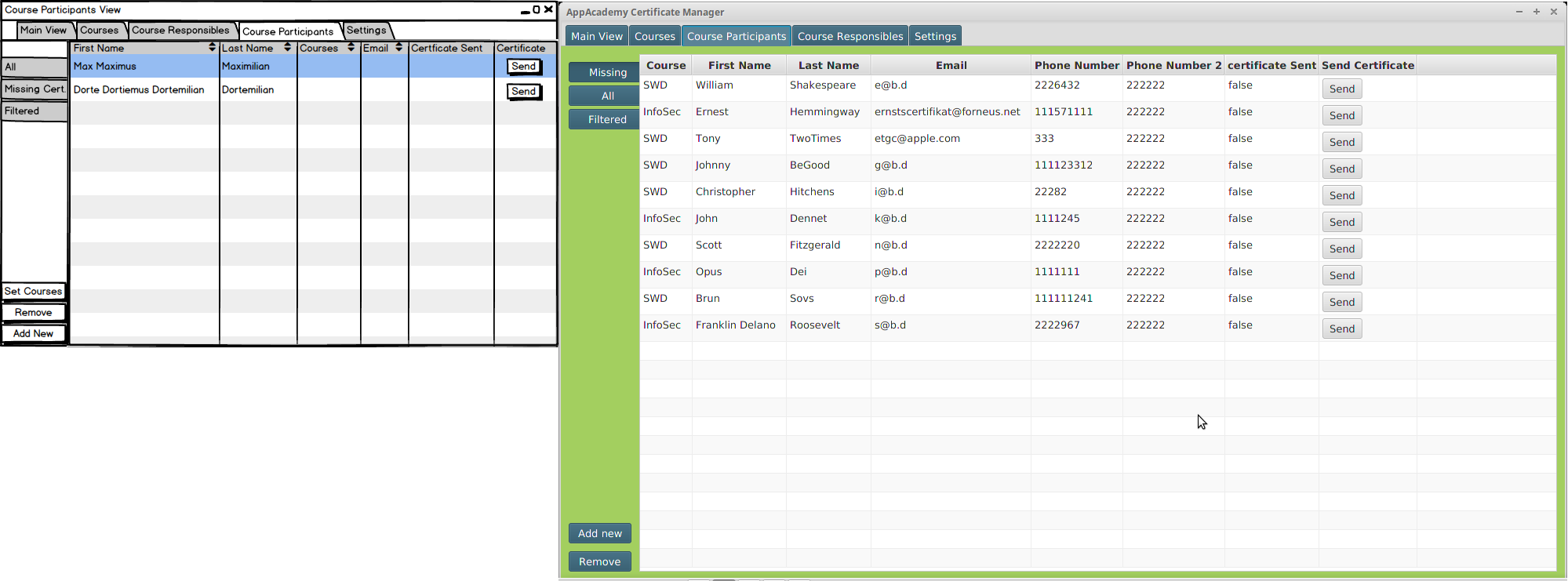


#### Main View

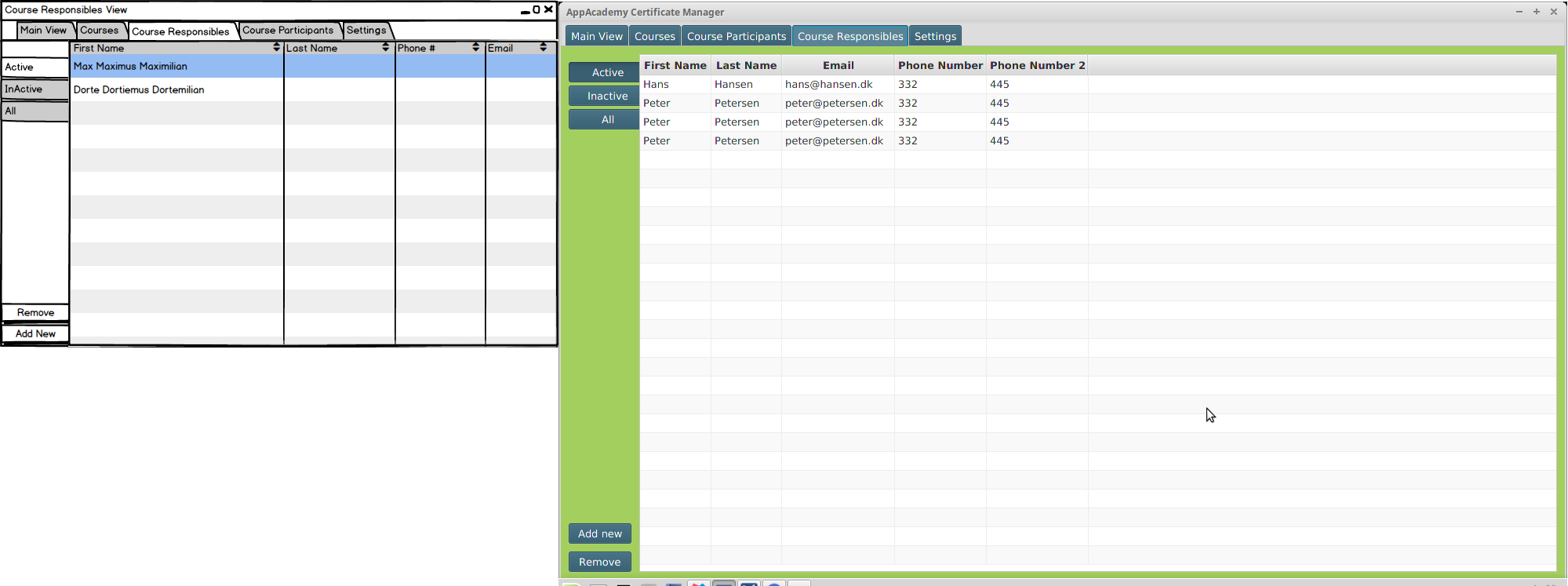


Courses View

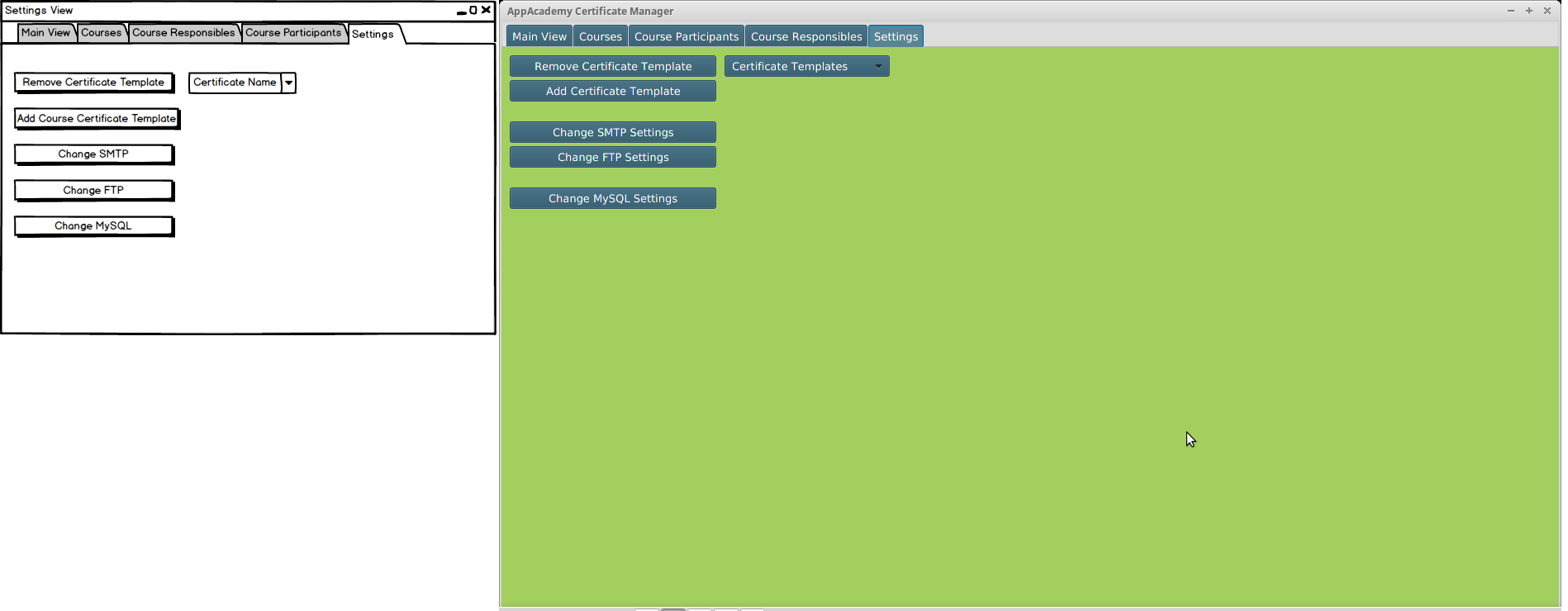
#### Course Participants View



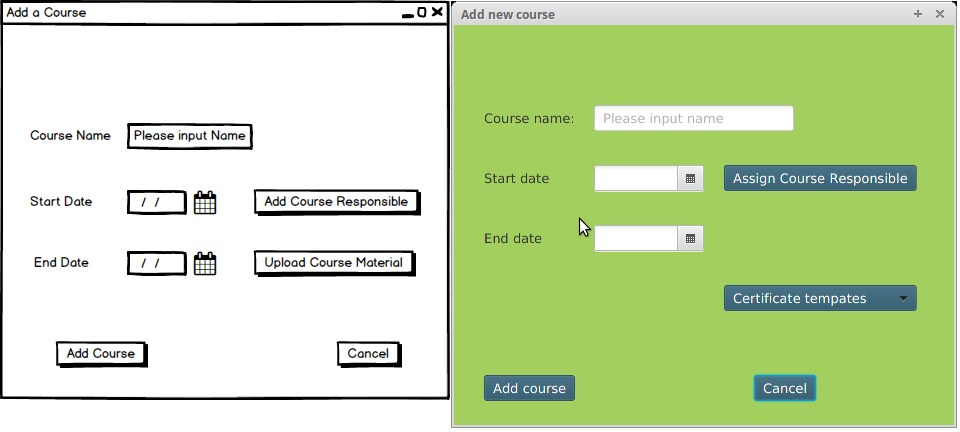
#### Course Responsibles View



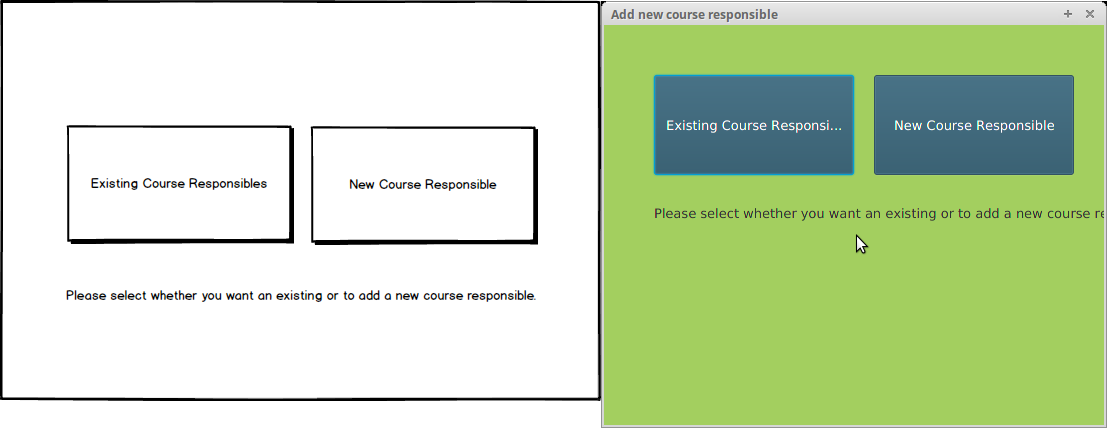
#### Settings View



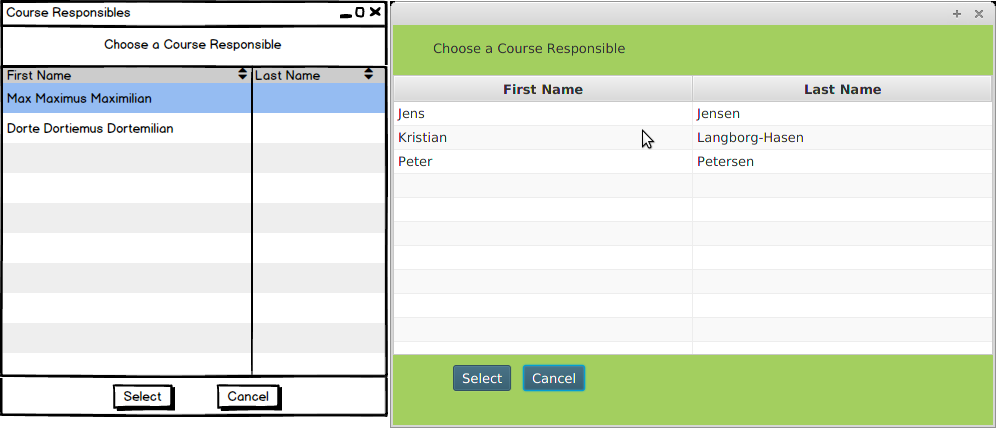
#### Add New Course



#### Select Course Responsible for Course



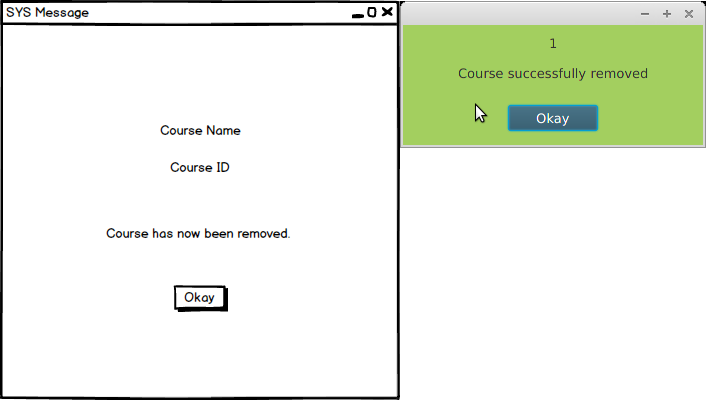
#### Select Existing Course Responsible for Course



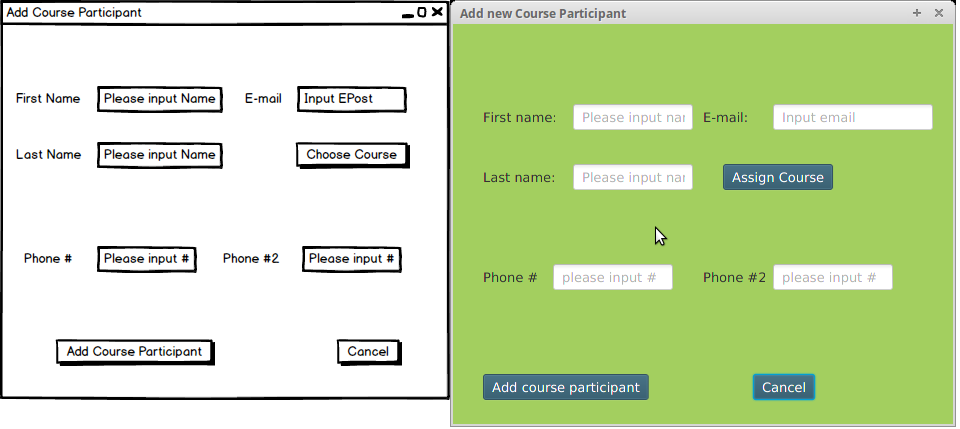
#### Remove Course



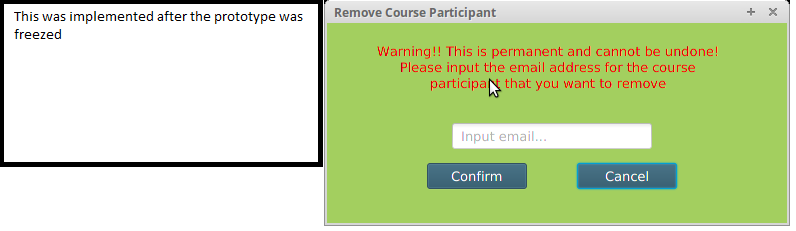
#### Course Removed



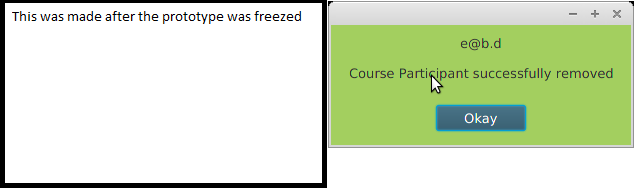
#### Add New Course Participant

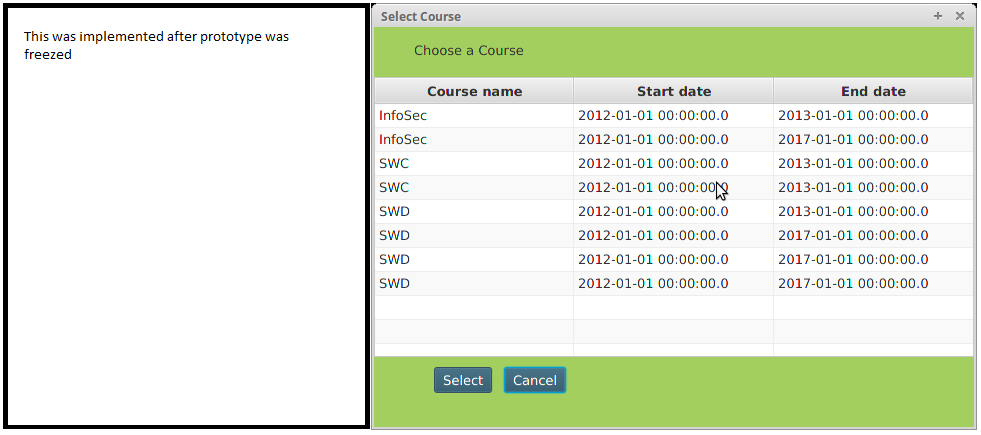


#### Remove Course Participant

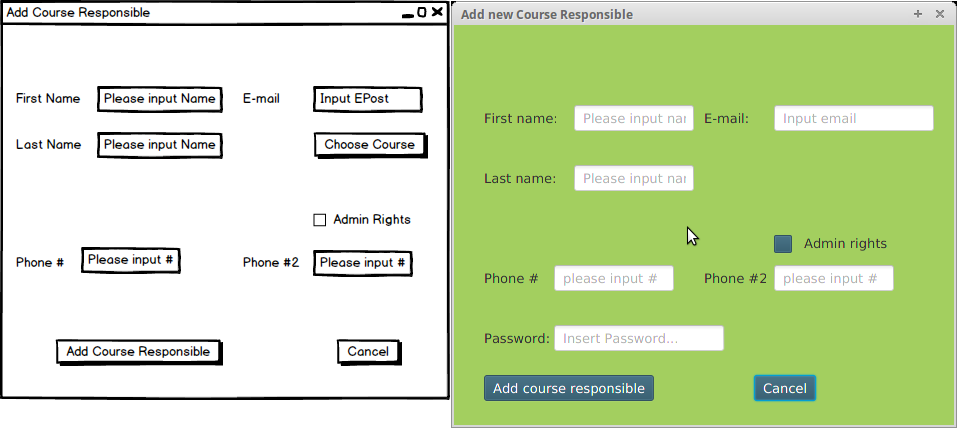


#### Course Participant Removed

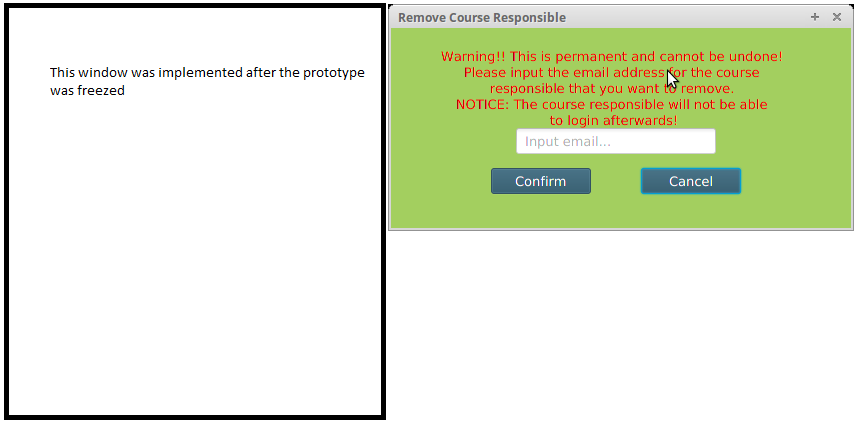


Assign Course Participant to Course

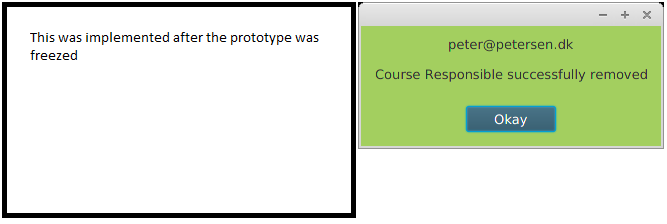
#### Add New Course Responsible



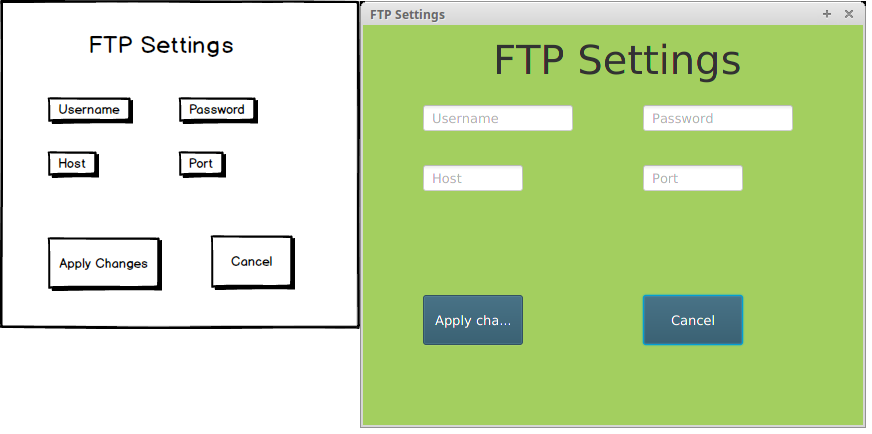
#### Remove Course Responsible



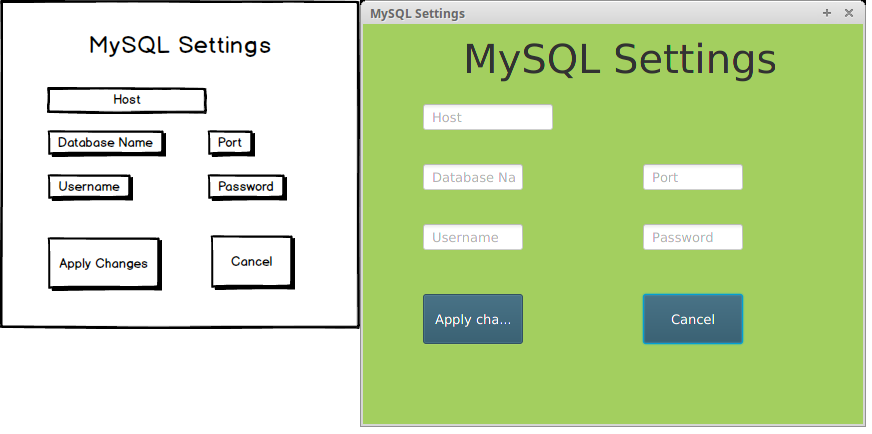
#### Course Responsible Removed



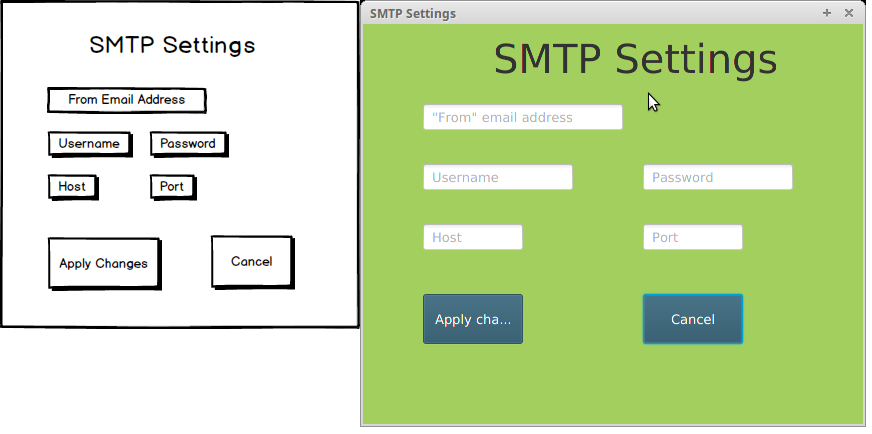
#### Change FTP Settings



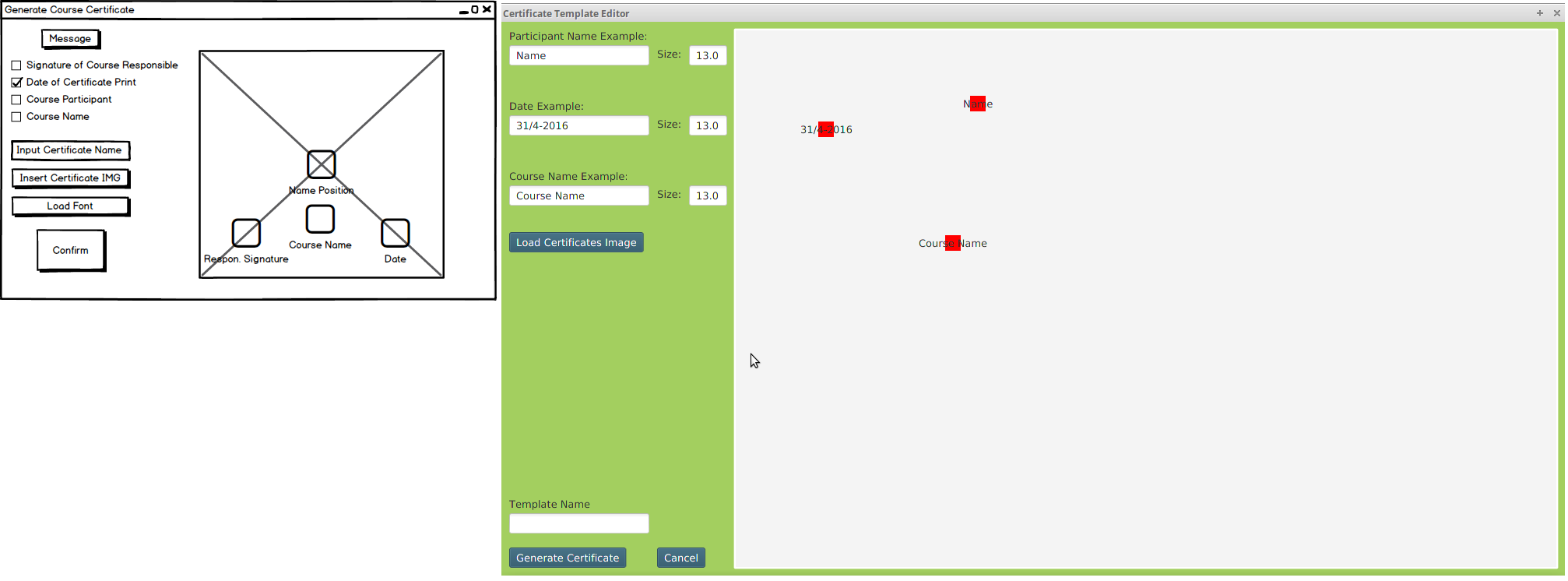
#### Change MySQL Settings



#### Change SMTP Settings



#### Generate Certificate Template



# OSCA

### OSCA Threading

In OSCA (Operative Systems and Computer Architecture) we were assigned with a task about threads.

We were given two java classes from our OSCA teacher Arne written with Swing instead of JavaFX. The classes has the screensaver's code except it did not quite work as intended, since the ball elements would randomly update and make the result look like many ball being frozen in the pane instead of the balls bouncing around as intended.

The classes consisted of a ball object class designed to setup moving GUI balls and a main class consisting of the main method. The assignment was about making a screensaver consisting of bouncing balls using threads as well as writing about the solution we came up with plus

another different solution to the problem as there are many.

#### Synchronized

The synchronized keyword ensures that a thread is using the correct data.

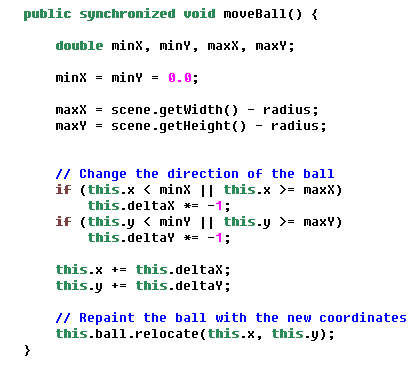
It also ensures that only one thread out of multiple can have access to a datasource at a time.

Using Java's 'synchronized' is used to wrap around the critical section to ensure that no other thread can access that part.

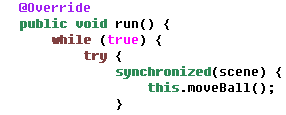
In the assignment, critical parts in the code are the scene element where the screensaver is supposed to run and the ball elements which are supposed to run within the scene. It is therefore very important that this scene is not altered by another thread.

To avoid this possible problem, we use the Synchronized keyword:

The moveBall method only has synchronized in it's method declaration which basically tells that all in the method is critical.



The run method settles for wrapping the scene inside a synchronized method within the run method's scope as the scene is the critical part here.



This upgrades the code we received to have the intended effect thereby solves the first part of the assignment.

#### Semaphore

A semaphore is a simple variable that says whether or not a resource is available. A thread asks for the semaphore to be locked if it isn't already used by another

thread. When a thread is finished in the critical region the semaphore is unlocked. Semaphores use a queue to avoid starvation.

The process for checking and changing the value needs to be atomic or else another thread could get the resource resulting in a deadlock.

# Summary

- Reminder: Husk at evaluerer hvordan vi glemte at implementere “Send certifikat” delen samt at Kristian også glemte at det skulle være med, da vi viste ham prototype 1.

# Bibliography

# Appendices

1. Course Material: Materiale\_Feasibility\_Study\_copy, Feasibility\_Studies\_IT\_af\_Steve\_McConnell, Feasibility Studies Hoffer 8 sider [↑](#footnote-ref-1)
2. Appendices: 1, 2 [↑](#footnote-ref-2)
3. <https://docs.oracle.com/javase/8/docs/technotes/guides/#javavm> / Visited 19.04.2016 [↑](#footnote-ref-3)
4. Appendices: 3, 4 [↑](#footnote-ref-4)
5. Course Material: Materiale\_Feasibility\_Study\_copy, Feasibility\_Studies\_IT\_af\_Steve\_McConnell, Feasibility Studies Hoffer 8 sider [↑](#footnote-ref-5)
6. Captains Log [↑](#footnote-ref-6)
7. <http://nyheder.tv2.dk/2014-12-19-tjek-her-laegger-du-dig-mere-syg-end-gennemsnittet> / Visited 19.04.16 [↑](#footnote-ref-7)
8. projekt\_og\_analyseredskaber.pdf / Figur 4.6 [↑](#footnote-ref-8)
9. Why Projects Succeed proactive risk strategi / Inspiration [↑](#footnote-ref-9)
10. See Appendix 5 for previous Iteration. [↑](#footnote-ref-10)