

Report- aCerts



School: KEA, Lygten 37, 2400 - København NV

Line: Datamatiker, 1st year Exam, 2nd Semester

Class: DAT15B

Group: 6

Date: 31.05.2016

Members:

Jonas Tvede Henriksen, 260495

Emil Lagoni, 051290

Dennis Hvidberg Madsen, 051093

Fredrik Constantin Schlander Sand Bjørklund, 080787

Table of Contents

Table of Contents	2
Introduction	4
Problem Definition	4
Company Description	5
AppAcademy's own Company Vision	5
How-To: Get Started.....	5
Software Requirements	5
Hardware Requirements	5
aCerts Dummy Users Login Information.....	6
MySQL Server	6
Server Information	6
GUI (Graphical User Interface).....	7
Gestalt Laws.....	7
Project Log – “Captains Log”	9
UP – Unified Process.....	19
Inception.....	19
Organization Type.....	19
Strategic Concerns Regarding implementation of aCerts.....	19
SWOT	20
Business Case.....	22
Vision for aCerts	22
Stakeholder Analysis.....	22
Feasibility Studies	26
Risk Assessment.....	29
Elaboration	30
Patterns	30
FURPS+.....	33
Use Cases.....	35
System Sequence Diagrams.....	45
Use Case Diagram	53
Domain Model.....	54
Sequence Diagram.....	55
Construction	56
Class Diagrams.....	56
Database	60
Entity-Relationship Diagram	60
Database Normalization Process	61
Database Integrity	64
Transition.....	65
Reviews.....	65
Program User-Manual / AppAcademy Certificates Manager	65
Cancelled Features	67
aCerts – Prototype contra actual Program	68
OSCA.....	77

OSCA Threading	77
Summary.....	79
Bibliography.....	80
Bibliography Code	81
Appendices	81

Introduction

Hello,

We're a 4-man group:

Dennis Madsen,
Jonas Tvede Henriksen,
Emil Lagoni,
Fredrik Bjørklund

Our approach has been a bit none-traditional. We are working together with a small course company called AppAcademy. We've decided that the way we would tackle this assignment would be through 100% group effort. This means that we decided to work 4 people on the same thing using a large screen projector. This of course has meant that certain things have been slower in the making, but also that we've had a combined knowledge pooled towards the same goals. This is the reason there won't be any specifically highlighted parts of the assignment as to whom made what, since we all made everything together and thus each of us is responsible for everything in this assignment, claiming otherwise would be falsification.

To maintain an overview of our sessions, we've rigorously kept a logbook, dubbed "Captains Log". We recommend reading it since it gives a tremendous insight into our work, progress and goals.

Our program is called aCerts, which is a course management tool that can send personalized course certificates to course participants. It will be a tool for internal use only. We hope you will enjoy our work.

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 ID.
- 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

AppAcademy's own Company Vision¹

“In the early days, computers were only for people specially trained in giving them commands. Today computers are everywhere: in smart phones, cars, televisions, medical equipment and much more. They are now so easy to operate that we don’t even have to understand how they work.

That is why many people today only learn how to use the digital devices, because they don’t think it’s necessary to understand them.

However, if you do not understand how a computer works, you can’t master it. You are deprived of your ability to innovate and limited to solving the tasks others taught the computer, rather than mastering the machine yourself, and using it to solve your own problems.

This is why more people need to learn programming.

It is not about making everybody software developers. It is about the fact that programming today is as fundamental a skill as reading, writing and math. It is a prerequisite for taking advantage of the possibilities of the digitalization – privately, during education, and at work.

If we do not change the current development, in a few years we will see the population split in technological A and B teams, where a small group of people with technological skills decides which possibilities to give to the majority of people.

At App Academy we think everybody deserves the ability to program, and we would like to help you learn it.”

How-To: Get Started

Software Requirements

(Libraries provided with Hand-in of assignment)

- Java 8
- Library: [Commons Net 3.5](#)
- Library: [javamail 1.4.5](#)
- Library: [mysql-connector-java-5.1.39-bin](#)

Hardware Requirements

- SMTP Server
- FTP Server
- MySQL 5.7.11, Critical not to use 5.7.12 (Newer versions cause crashes / localhost/127.0.0.1)

Program is located within aCerts Folder along with a dummy database.

¹ This is AppAcademy's own material, not produced by us.

aCerts Dummy Users Login Information

Admin

User: klh@appacademy.dk

Password: 1

None-Admin

User: jens@jensen.dk

Password: 1

MySQL Server

User: root

Password: 12345678

This information can be changed within the source code. See file SceneInitializer, Line 34.

Remember to run “create_database.sql” file, which contains a dummy database to try things with.

Server Information

(Active for duration of Assignment)

SMTP

Username > certifikat@forneus.net

Password > password

host> send.one.com

Sender Email> certifikat@forneus.net

Port> 587

FTP Administration

Host: [ftp.forneus.net](ftp://ftp.forneus.net)

Username: forneus.net

password: emildennisjonasfredrik

Port: 21

GitHub

Url: <https://github.com/ChuckDK/examProject>

Repository will be made publicly available June the 1st 2016 at 12:00 pm.

GUI (Graphical User Interface)

Gestalt Laws

Law of similarity

We have used the law of similarity by using color to group elements together.

For our program vi have been given some color codes from our client that he wanted us to use. We have used the color so that the same colors are used for the same type of elements throughout our program, to get some consistency in our program.

All of the buttons in the program have been given the color blue, so that the color blue is associated with elements that you can click on.

Green have been used as a background color and nothing else, so every time something is green (in this case only our background and the company logo) it is something that cannot be interacted with.

All kinds of text fields and tables have been given the color white with black text, to provide readability because of the very high color contrast.

Law of proximity

The law of proximity is used on multiple occasions in our program:

- Login screen

The text fields for inputting username and password are placed right above each other and have the same length and height to show that they are connected to each other.

The login button is placed right below the login text fields to show that it is also connected to the login info.

The exit button also uses this law but it is used to show that the button is NOT associated with the login info, by moving it far away from the elements associated with logging in. This shows that the exit button closes the program no matter what login info is provided.

- All the table views

In all our table views we have buttons to the left of that filters the information in the table views. These buttons are placed right above each other, to symbolize that they have the same kind of function (they are all filters)

the add and remove buttons are placed in the button far away from the filter buttons to show that their functionality is different.

- Settings view

In our settings view the buttons are grouped together position ally so that buttons concerning certificates are placed together, all "transfer protocol" settings are grouped (SMTP, FTP) and MySQL settings is placed alone at the bottom.

The certificate templates dropdown menu is placed right beside the remove template button to show that those 2 elements are grouped functionally.

[Law of closure](#)

In the certificate template editor view all buttons and text fields are grouped together in the green area to the left of the window that can edit properties of the elements in the editor. Everything that will appear on the finished certificate are grouped in the editor itself (the white area).

This law also applies to all the tables where all of the elements within the tables only interact with the rows they appear at.

Project 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 digitalized 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 realized 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: 121.5 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 scheduled 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 unforeseen closing of KEA, 6 hours SWK

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

ETA: 111.5 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: 101.5 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 Course Material 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: 91.5 Hrs. Remaining pr. Developer

May 3rd 2016

Today we identified processes according to their correct UP position. We finalized all our Use Cases and the Domain Model. We started Use Case Diagrams, but we haven't been able to finish this during todays session. FURPS+ has been finalized. 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: 81.5 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 finalized 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 an approach that risks being classified as Waterfall model and iterations will regain focus now.

Next meeting: See Progress Table
Finalize 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: 71.5 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 thorough 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 implemented 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: 61.5 Hrs. Remaining pr. Developer

May 13th 2016

Today we had our regular Friday crunch day. We finalized 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 assign our Course Participants to a Course, this will be corrected for next time. We still haven'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 budgeted 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 a lot more time assigned to SWD than will be used with concurrent ad-hoc tasks. We will agree on how much time should be taken from Buffer and maybe some from SWD as well during the next Meeting, 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: 51,5 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 finalizing 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: 41.5 Hrs. Remaining pr. Developer

May 20th 2016 / Iteration 3 ends

Today has been the end of Iteration 3. We've implemented a lot 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 as well, 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: 31.5 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: 21.5 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: 11.5 Hrs. Remaining pr. Developer

[May 31st 2016](#)

So today has been the final day of our project. It has been quite hectic, retrieving our sources for bibliography, finding the correct places to input these and also structuring so many hours of work into a single coherent report. We're all exhausted and the end of the project will be welcomed. We've all learned a lot during this project, many things that we would do differently and there is no doubt that we can work more efficiently come next large project. We've cleaned up our GitHub folders and we're ready to ship. It has been a pleasure, "but all good things must come to an end".

Members Present: Dennis, Fredrik, Jonas, Emil

Developer Hours: 10 hours Buffer

ITO: 14/20 – SWD: 33/70 – OSCA: 4/10 – SWK: 57/30 – Buffer 12/61.5

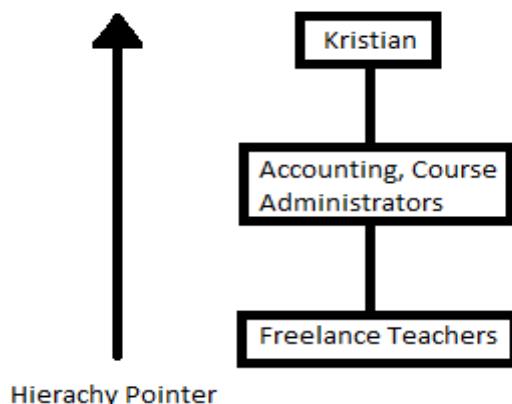
UP – Unified Process

Inception

Organization Type

AppAcademy is a small it-courses organization, it counts less than 10 full-time employees and uses a lot of external lecturers for its courses. Its primary focus is towards the public school system in Denmark and at a national scope. As a company it has a flat structure without an apparent hierarchy, Kristian(CEO) is into an open-door policy type of leadership. Yet we can provide a tree overview of the company's hierarchy, we find it unnecessary with a company of this size but have decided for it anyway as it's impression that it's expected as part of the assignment.

Hierarchy of AppAcademy



Strategic Concerns Regarding implementation of aCerts

As it stands right now Certificate Management for the Courses is handled manually by a single employee at AppAcademy. This employee manually customizes each certificate in Photoshop one certificate at a time. Our program will make her work process, extremely efficient by comparison, to such an extend even as per conversation with Kristian, make her job obsolete. He has mentioned that he will be looking for new tasks to do at AppAcademy since there will be a lot of freed time from her schedule.

It will also give him a better overview of courses and certificate handling in general. A scenario he suggested that wouldn't have been possible previously is that he now would be able when having a customer on the phone to immediately expedite a certificate through the system and also see if it's been send at a previous point in time. All this is in such a short timeframe to receive

immediate confirmation if said certificate has been received via phone. This entire process was a lot more demanding previously.

SWOT (m.fl., 2012)

Strengths	Weaknesses
<ul style="list-style-type: none"> AppAcademy has been around for some time which has provided them with experience and tons of cases from which they can structure courses. This also provides them with an advantage when it comes to sales. AppAcademy has 2 different types of competitors in the courses market. Companies that hold courses outside office hours or during office hours as is the case with AppAcademy. Several competitors don't have courses as primary business and treat it as a source to a secondary income, whereas AppAcademy does this as a full-time job, which means their approach in general pertains a more serious character. AppAcademy also is 100% self financed, there are no 3rd parties involved. They are independent from bankers and investors, which provides them with the means to make quick decisions and frees them from bureaucracy. Kristian(CEO), also has a mentor with a high level of expertise who consults with him on a regular basis. This of course also brings some added experience to the table. 	<ul style="list-style-type: none"> AppAcademy needs to strengthen its Marketing department, the ROIs (Return on Investment) could be better. There is a huge time investment into each customer before an order for a course has been secured, this needs to be improved. It's hard for the company to allocate resources to put together a new type of course. It's mostly based on estimates and if these aren't in sync with customer demand, they won't make any money on them. This is a high risk part of the business. Quality insurance, it's very expensive for a small enterprise to have quality insurance compared to larger companies, AppAcademy has to do high-cost management tasks for a small budget company. This is a definite downside by being decidedly small. Lack of quality insurance, such as bad Course Responsibles can result in lost recurring revenue of courses.

Opportunities	Threats
<ul style="list-style-type: none"> Several opportunities can be scouted on the horizon. Schools currently have a renewed focus on IT and programming in general. AppAcademy has invested into being able to handle the large flow of High School teachers that will need new qualifications to handle these new demands. Currently the government is negotiating a school reform that in its current form seeks to make programming mandatory for students. There is a huge surge in online video courses, a large part of national competitors have spent huge sums of money on physical locations to have their courses. This is no longer an advantage with the market changing, AppAcademy on the other hand has primarily focused on IT-infrastructure as their primary source of investment, which has them prepared to meet the current market change head on. 	<ul style="list-style-type: none"> AppAcademy's biggest threat currently comes in the form of a government organ called CFU (Center for Undervisningsmidler). CFU used to handle school libraries and resources for these, since there is no large demand for this any longer, they are considering themselves to start providing courses to the schools as a new service at a price point where AppAcademy can't remain profitable. They won't be competing on market terms. Online Video Courses such as Lynda & Coursera are a threat, even though online video courses seen on a local/national scale is an opportunity. So here we have an overlap where a market change can be perceived simultaneously as a threat and an opportunity at the same time. The company's size makes it vulnerable to personnel changes. Every head count is essential to the company. Sales have to perform. AppAcademy's situation as a financially independent company of its size means cash flow has to be present, since capital is a scarcity and the company could only remain afloat for a couple of months if sales were to stagnate.

Business Case

Vision for aCerts

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

Stakeholder Analysis (m.fl, 2012)

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

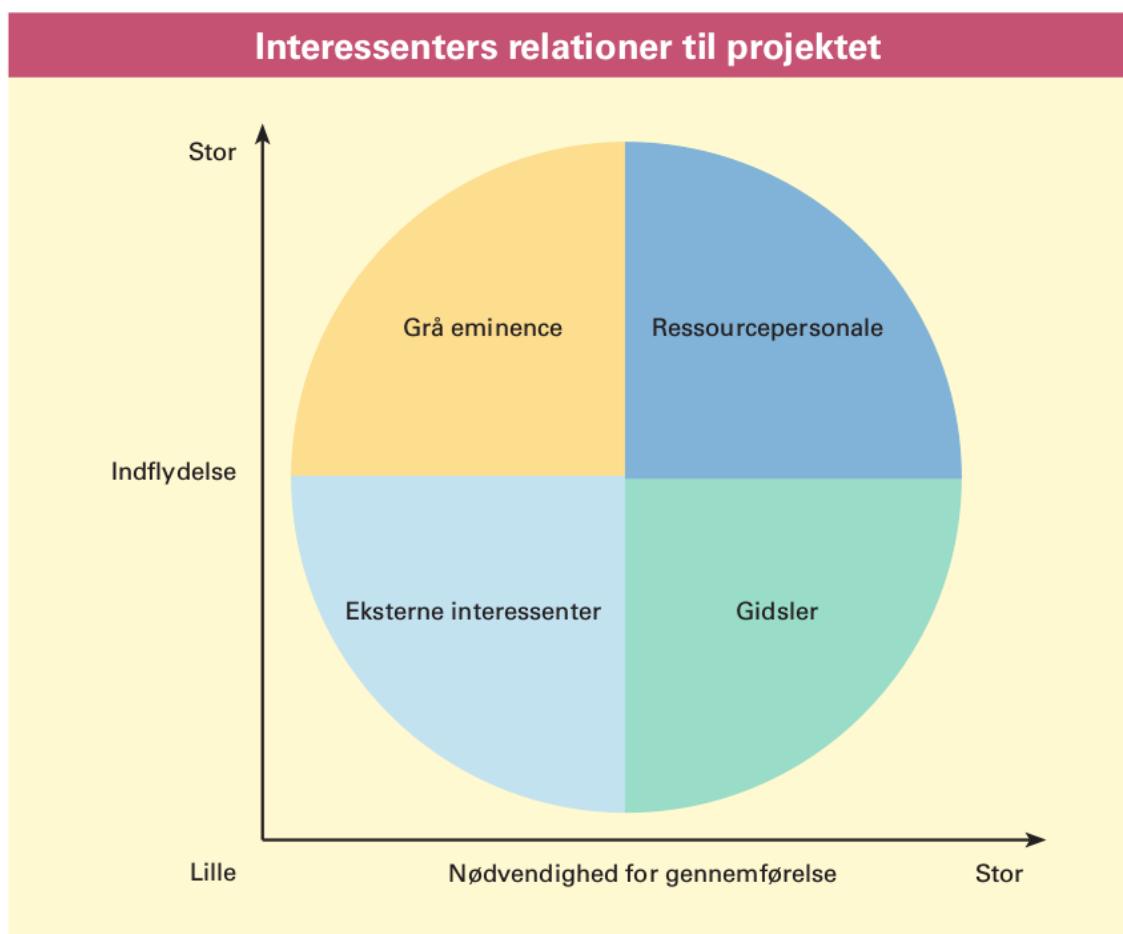


Fig. 12.7 Interessenter opdelt efter indflydelse og nødvendighed for gennemførelse.
Kilde: Inspireret af O.Andersen, N. Ahrengot & J.Olsson: Aktiv projektledelse, Børsen, 1999.

Grey Eminence: KEA

Hostage/Resource Stakeholder: Kristian

Hostage: Course Responsibles

Resource Stakeholder: 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 overrule 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/Resource Stakeholder: 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 terms 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 realized.

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 (Ema, 2016)

Technical Feasibility Study² (Steve McConnell, n.d.)

- 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 (Oracle, n.d.).

² Appendices: 1, 2

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 hindrance for us to proceed with the project. We have other solutions ready if needed.

Schedule Feasibility Study³ (Steve McConnell, n.d.)

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

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 spent 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 (TV2, n.d.) 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% 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:

³ Appendices: 3, 4

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 (Anon., n.d.)

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 Continuously 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 (Kastner, 2011)

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. Therefore 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

Patterns

[Grasp \(Larman, n.d.\)](#)

To create the program, we have used some of the patterns which falls under the General Responsibility Assignment Software Patterns (GRASP).

The GRASP patterns used are the Controller, low coupling (as much as possible), High Cohesion, and Polymorphism patterns.

In the following we will give a brief description of each pattern and an explanation of how we have used each pattern and why:

[Controller](#)

The controller pattern dictates how there should be classes which handles requests from the UI, making it so the UI essentially has no functionality itself.

We have actively used this principle throughout our program as we have a "Controller" folder in our program which our GUI sends user events to in order to get the results.

Low Coupling

This pattern dictates that the classes should have as little relation to other classes as possible, which essentially results in the following perks: lower dependency between the classes, changes in one class have lower impact on other classes, and higher reuse potential.

We have used the principles of this pattern as far as we are concerned meaning we have lowered the relation between classes as much as possible and have gone for high reuse potential e.g. our `FileIO` class which has one responsibility but can be reused. However, we could have had even lower coupling by having a factory class which creates some elements used in the method bodies from our `"MySQLParticipants"` class.

All in all, we designed the program with low coupling in mind but when we were finished programming our program was not as low coupled as we wished for.

High Cohesion

The High Cohesion pattern dictates that a class should have moderate responsibilities in one functional area and collaborate with other classes to fulfill tasks.

We have used this pattern by creating subsystems where each class use their own responsibility along with other classes in the same subsystem to create one output. An example of this in our program could be the `"operations"` and `"settings"` systems. In the `"settings"` system, the `FileIO` class read and writes objects which the `"FTPSettings"`, `"SMTPSettings"`, and `"MySQLSettings"` uses to be updated with and thereby set the settings in the program.

Other patterns

During the creation of the program we have also used some patterns which do not fall under the GRASP patterns;

Model View Controller (MVC) and Singleton patterns.

Model View Controller (MVC) (Larman, n.d.)

The MVC pattern is an architectural pattern which means it is a pattern used for organizing the structure of the program. The view is what the user interacts with and should be free from any functionality meaning it should call upon the controller to have the user events handled. The model is the objects where the handled information is `"packed"` and then shipped on to the view in order to update it, or save it to a file on the pc.

The components cooperate as follows: The user sees the view and for instance pushes a button which triggers an event. The view calls upon the controller which in turn manipulates the event and saves the result in the model which is sent to the view and thereby updates it for the user to see.

We have used this pattern to structure our program entirely; Our GUI elements are in the view folder, our functionality is in the Controller folder, and the objects we initialize any event triggered handled information is in the model folder.

Singleton (Larman, n.d.)

The Singleton pattern which helps creating one and only one instance of a class. The way it is implemented is by making the constructor in the class private in the class you wish to be based on singleton. Then in the same class you make a static method (usually called `getInstance`) which returns an instance of the class, but first checks if an instance has been created. If this is the case

the getInstance returns the already created instance meaning only one instance of the class will be present when you run the program.

We have used the singleton pattern in our ViewPanesManager class to ensure that you can't change your view to be that of an admin if you are a normal user. The singleton pattern ensures that the normal instance you logged in with is the same throughout the entire program session.

FURPS+ (Larman, 2004) (Bruegge & Dutoit, 2010)

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 has 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 sanitized and only current administrators can create other Course Responsibles.

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 no interference with Danish law about storing data that can identify people.

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

Use Cases (Larman, 2004)

1. 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 storables.

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 save.
 - 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).

2. 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 accidentally 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 sent.

[3. 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(eJava 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?

4. 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.

5. 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.

[6. 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. PortHe 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.
7. [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 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.
8. [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:
 - a. Username
 - b. Password
 - c. Host
 - d. Port
 - e. 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.

[9. 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'.

[10. 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.

[11. 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.

[12. 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.

[13. 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.

[14. 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.

[15. 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.

[16. 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.

[17. 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.

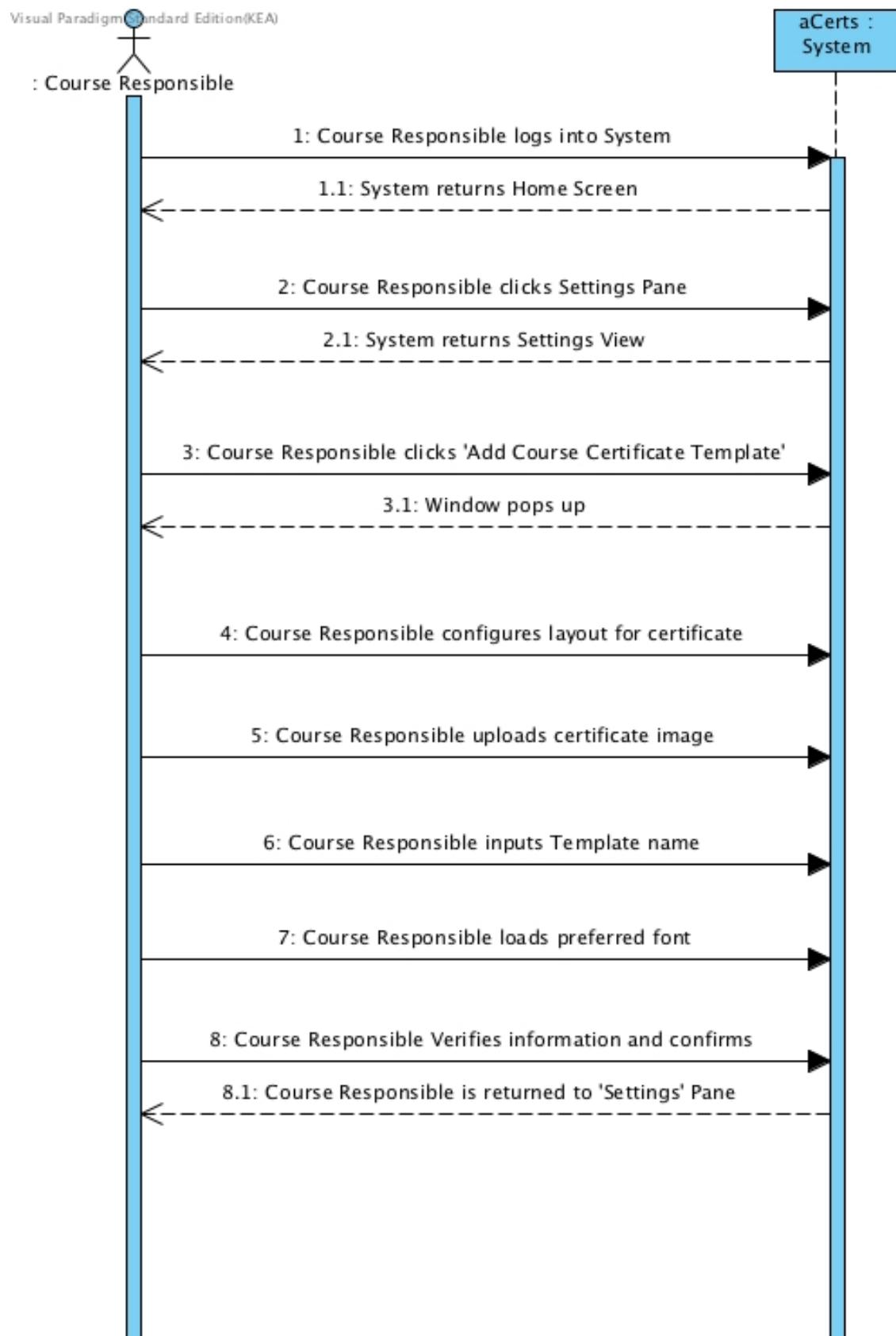
[18. Brief: View inactive Courses](#)

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.

[19. 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 (Larman, 2004)



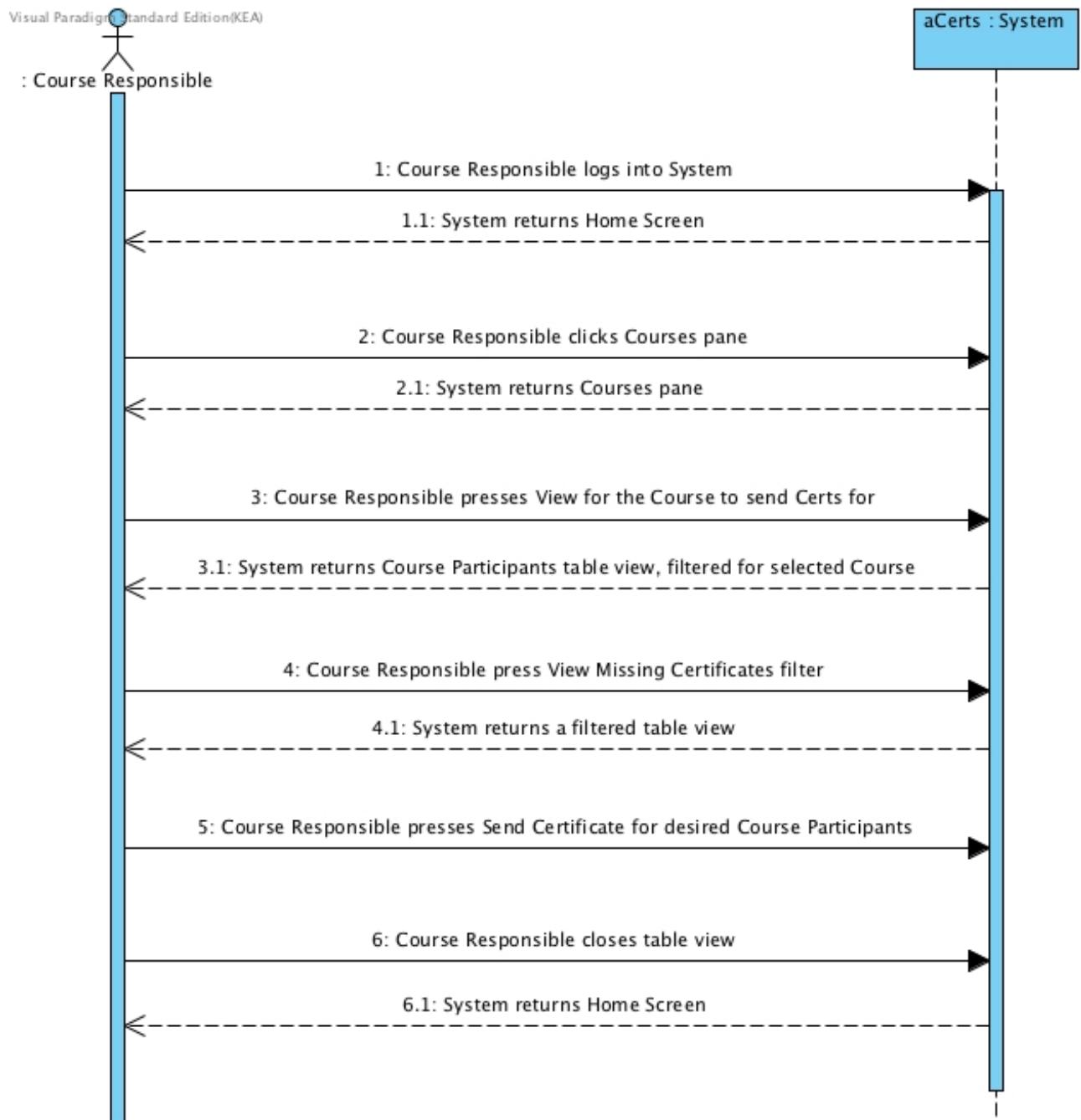


Figure 2. Send Missing Certificate

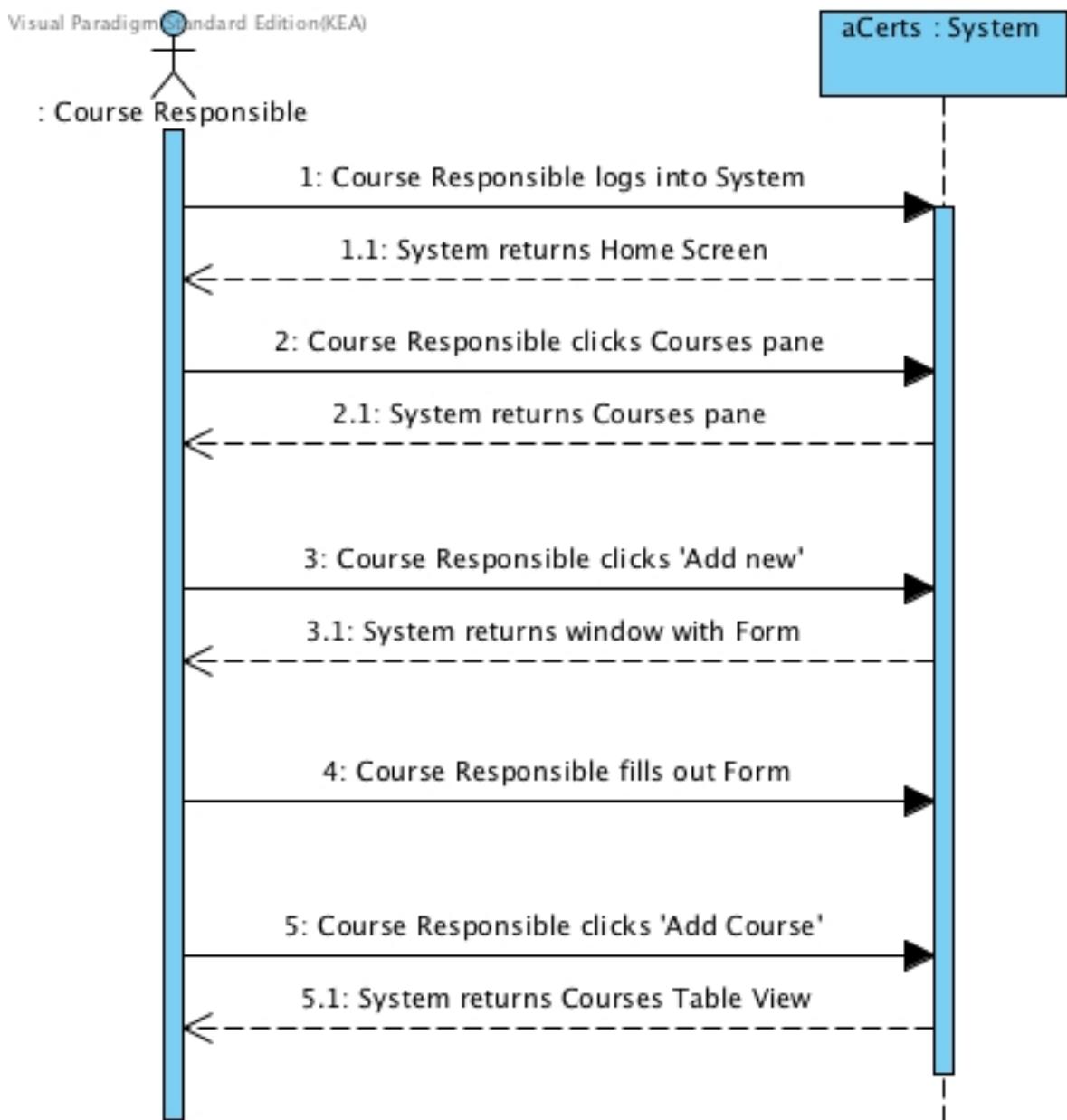


Figure 3. Add Course

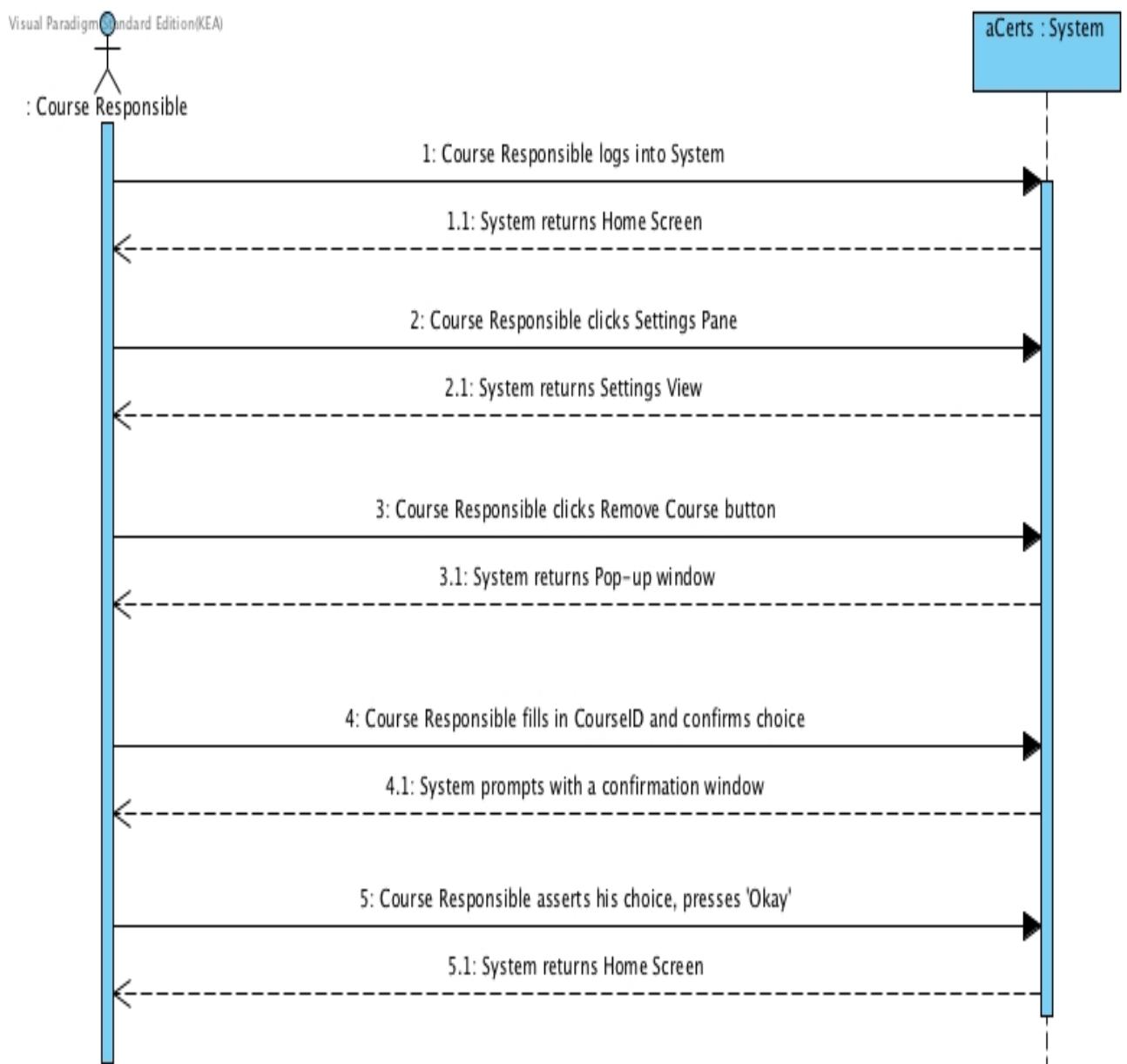


Figure 4. Remove Course

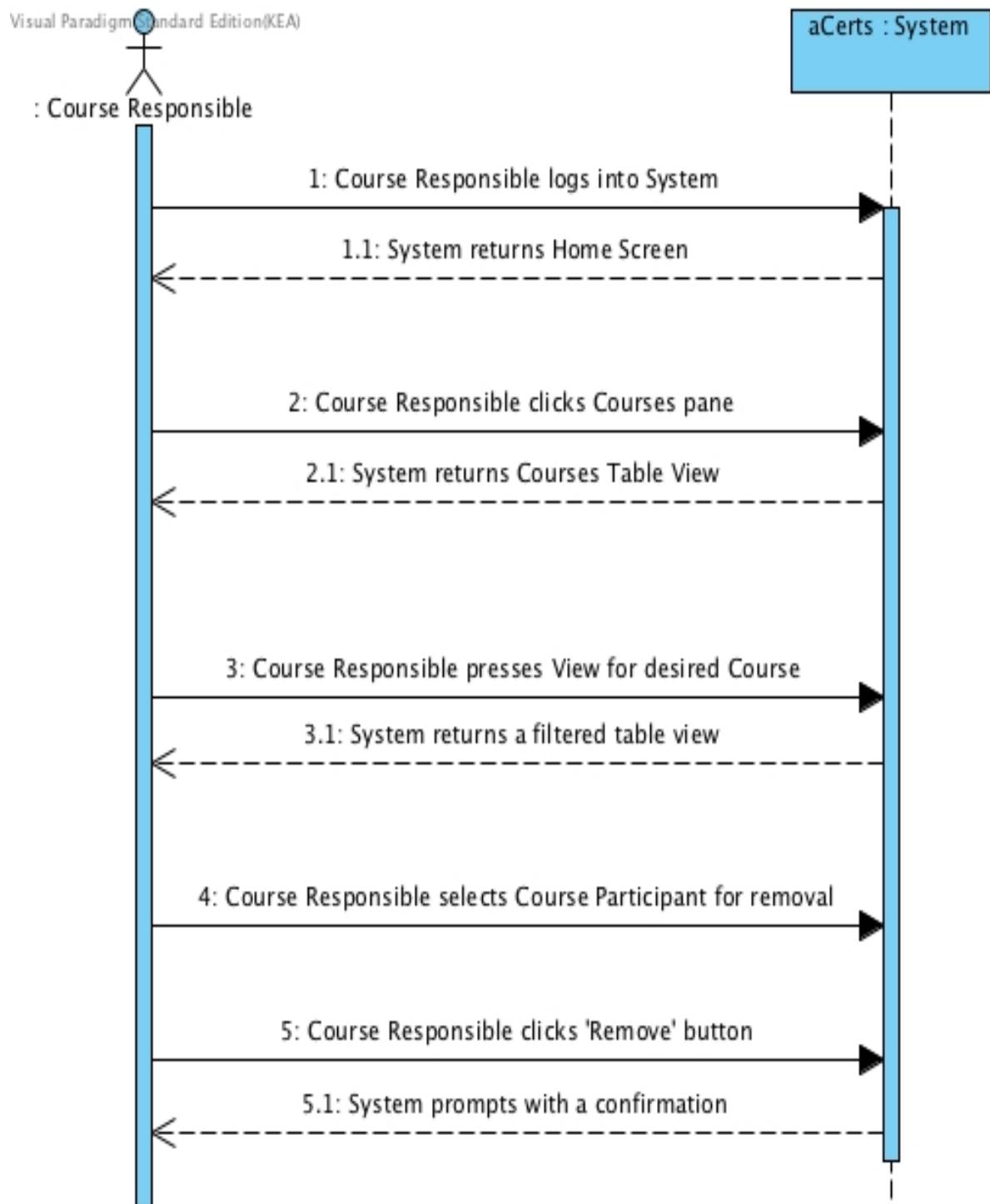


Figure 5 Remove Course Participant

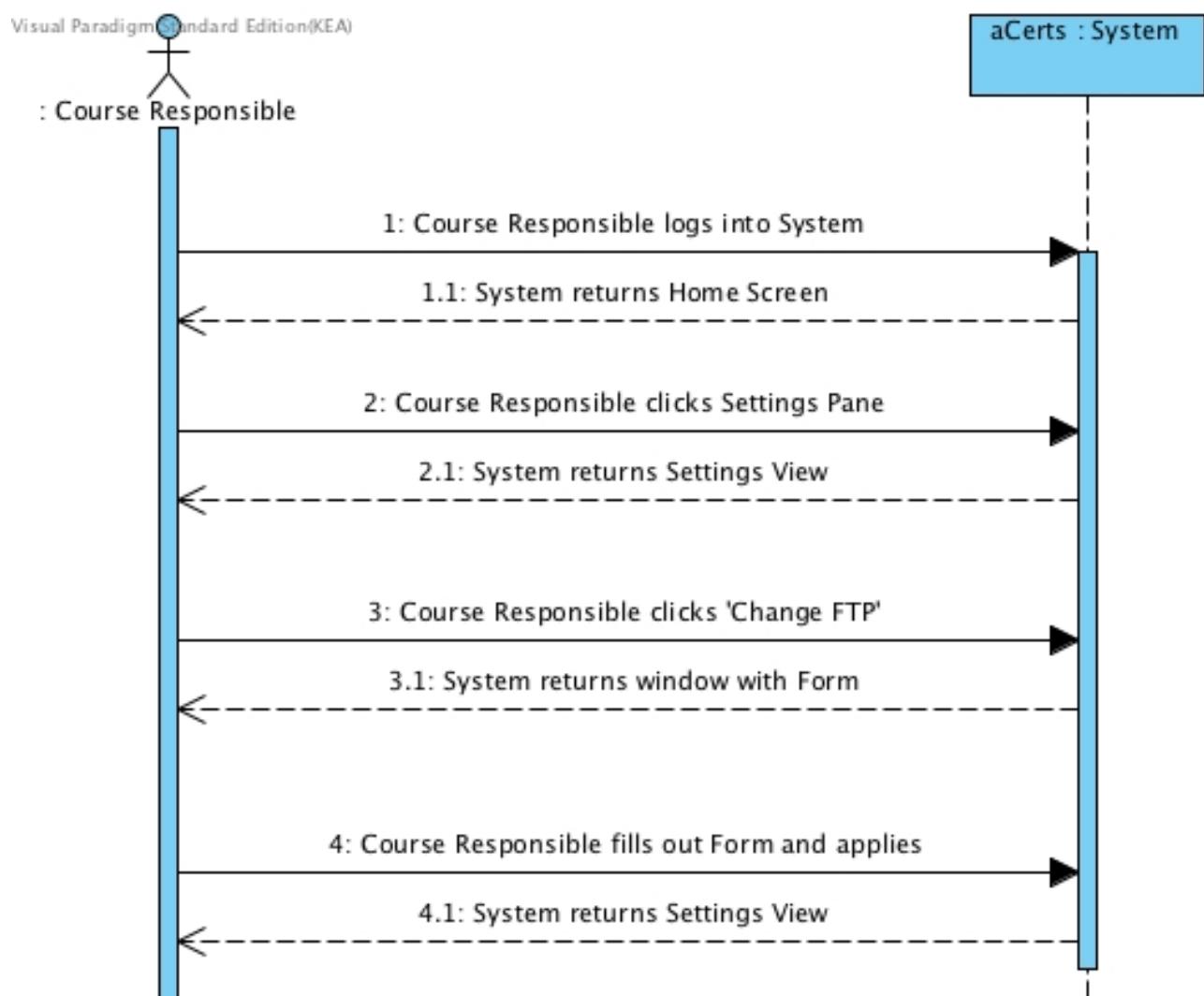


Figure 6. Change FTP Settings

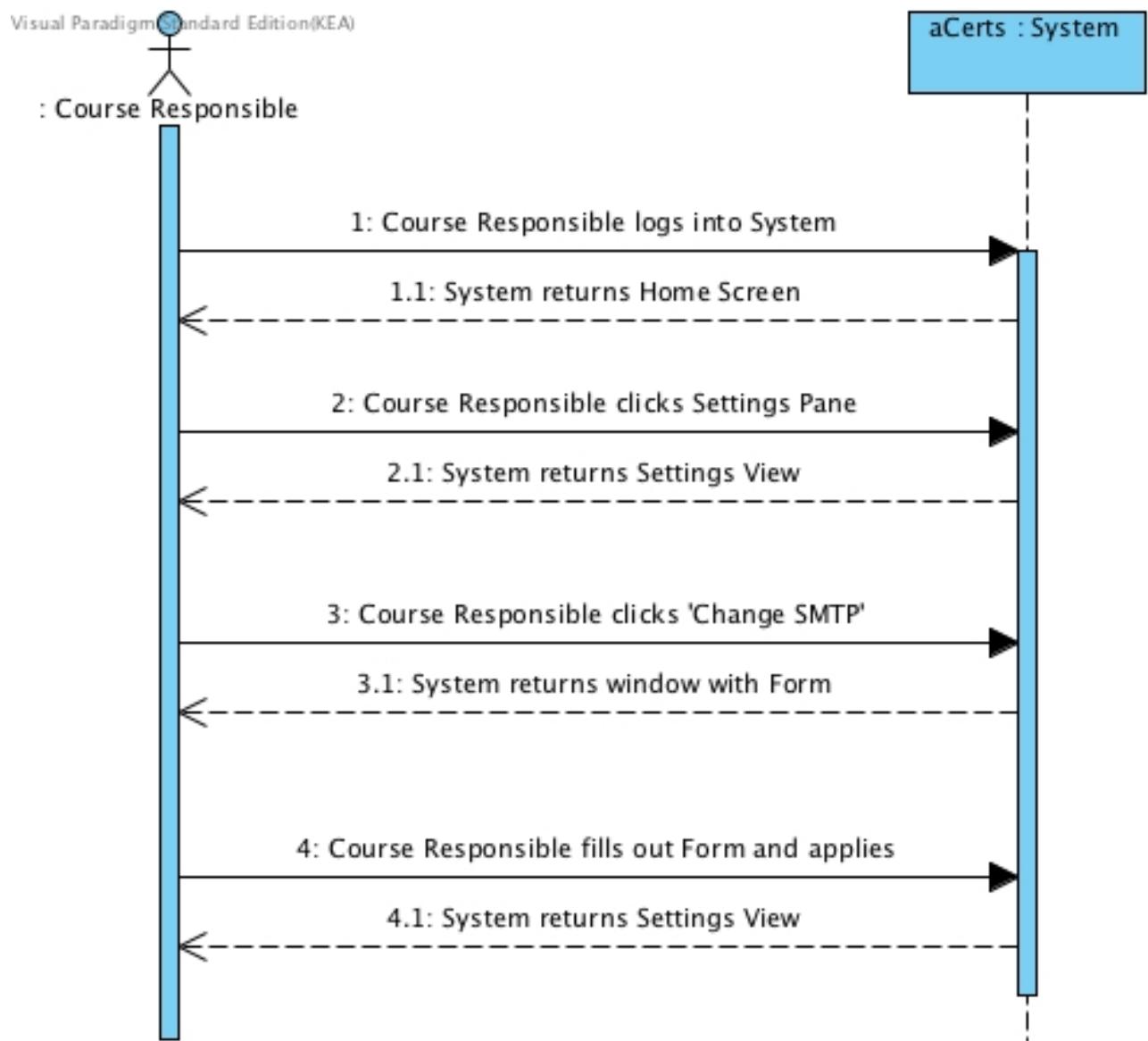


Figure 7. Change SMTP Settings

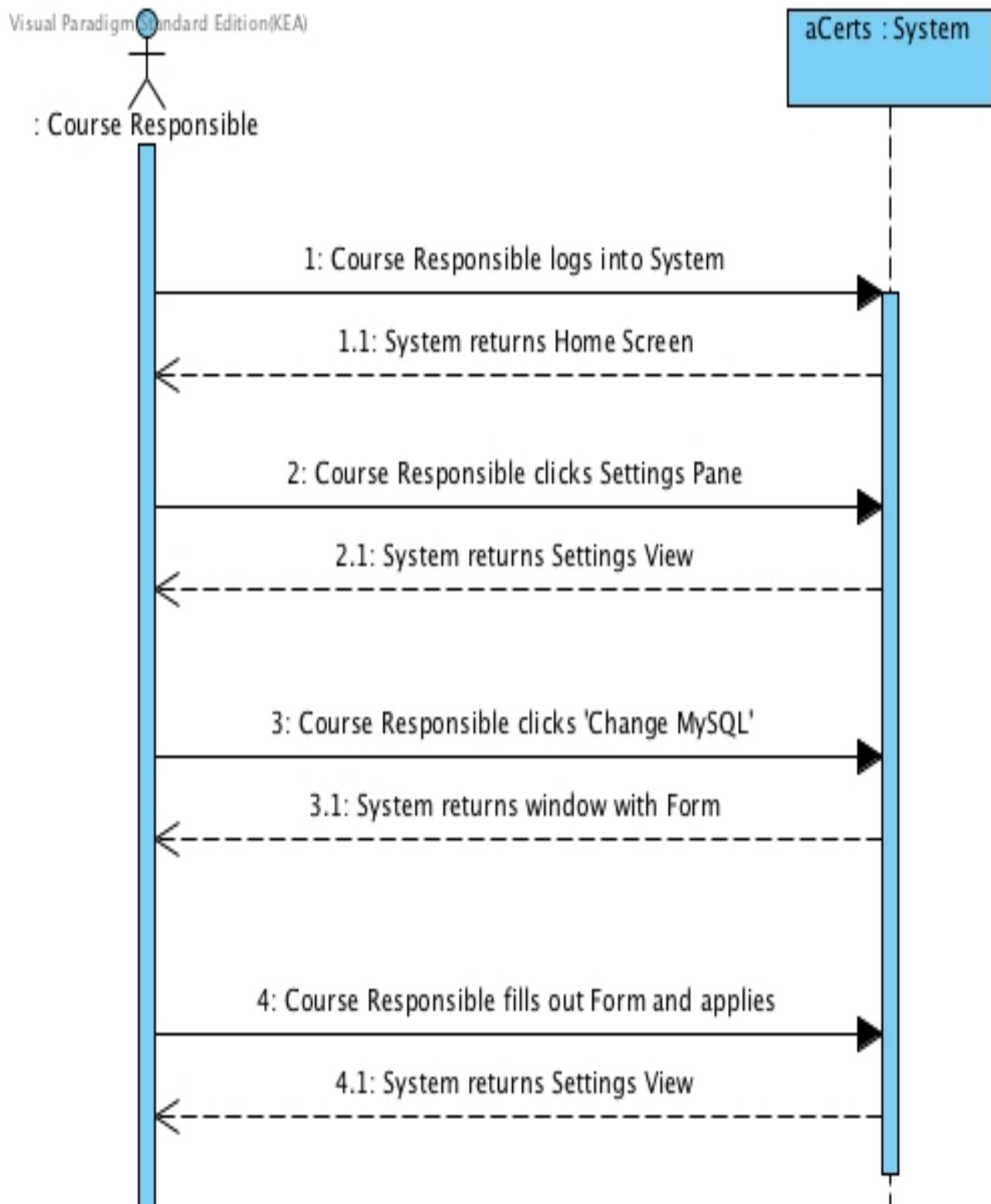
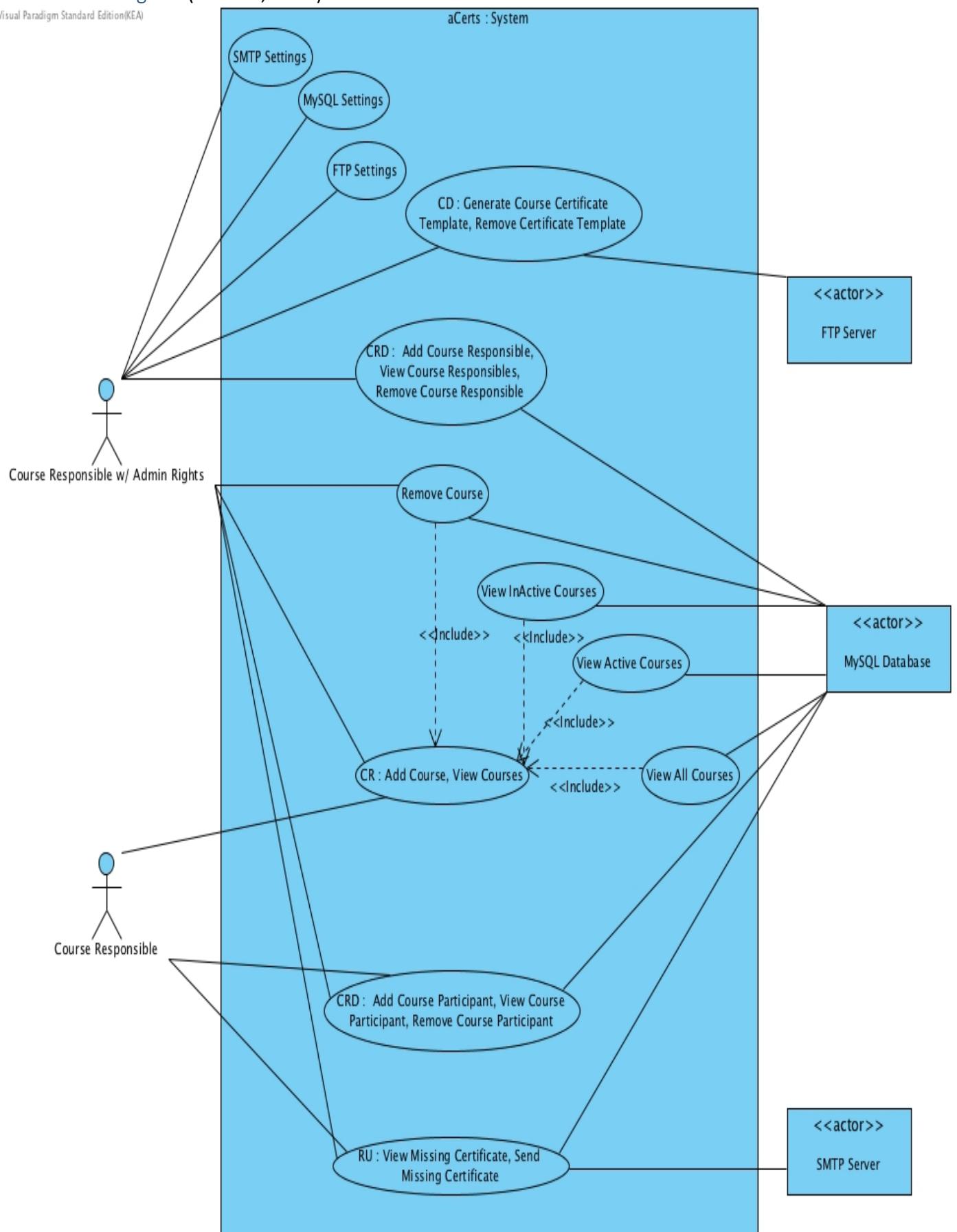


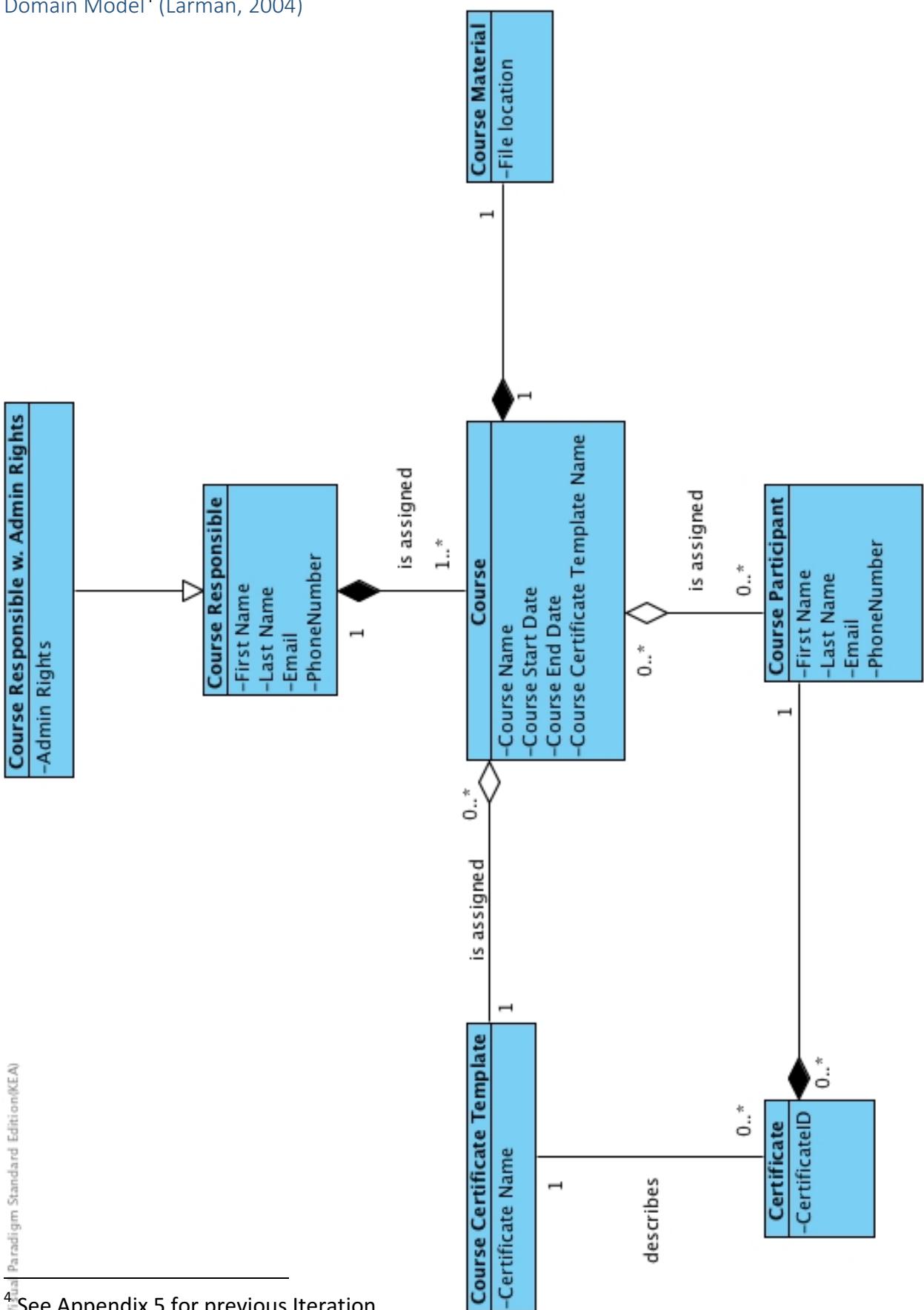
Figure 8. Change MySQL settings

Use Case Diagram (Larman, 2004)

Visual Paradigm Standard Edition(KEA)

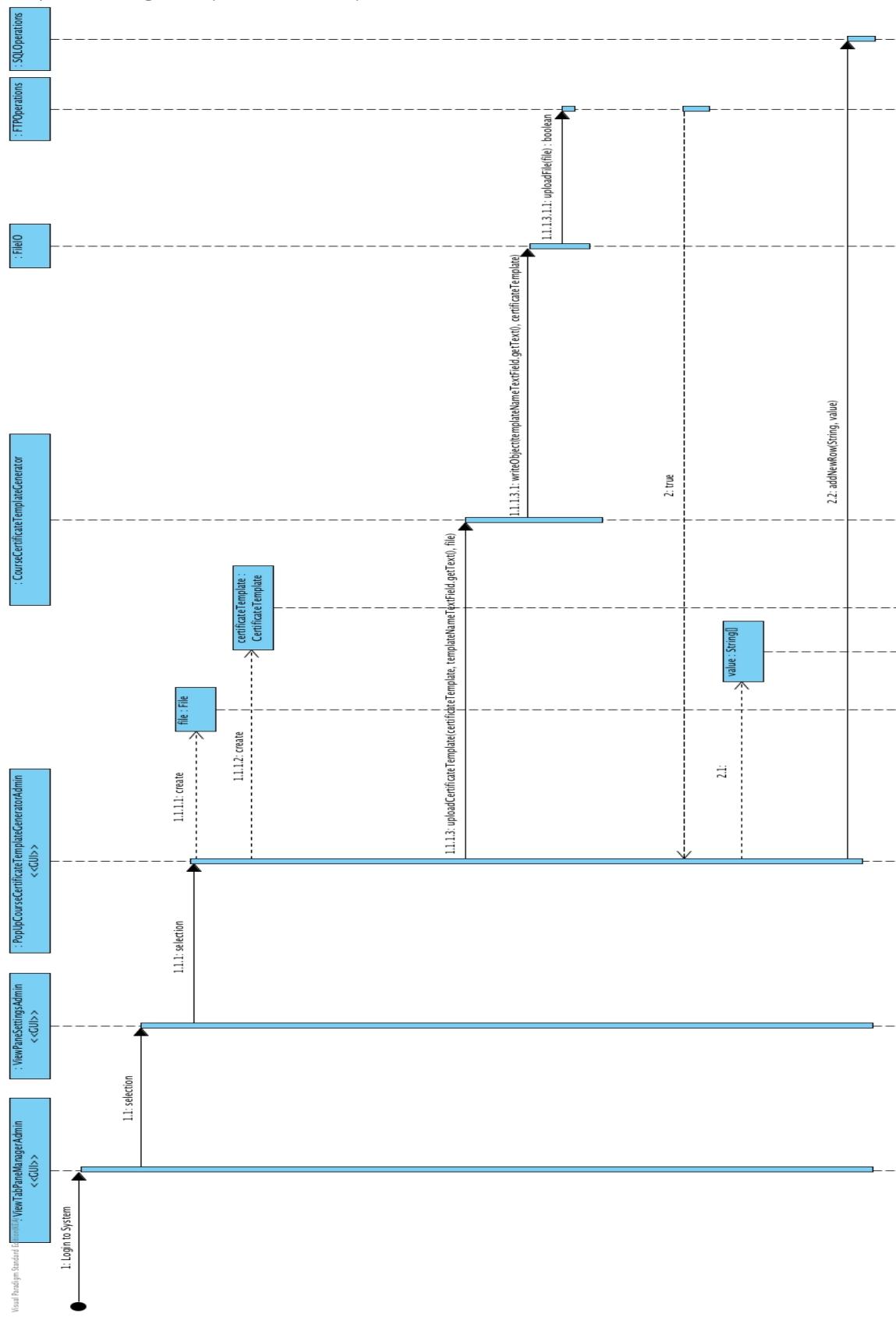


Domain Model⁴ (Larman, 2004)



⁴See Appendix 5 for previous iteration.

Sequence Diagram⁵ (Larman, 2004)

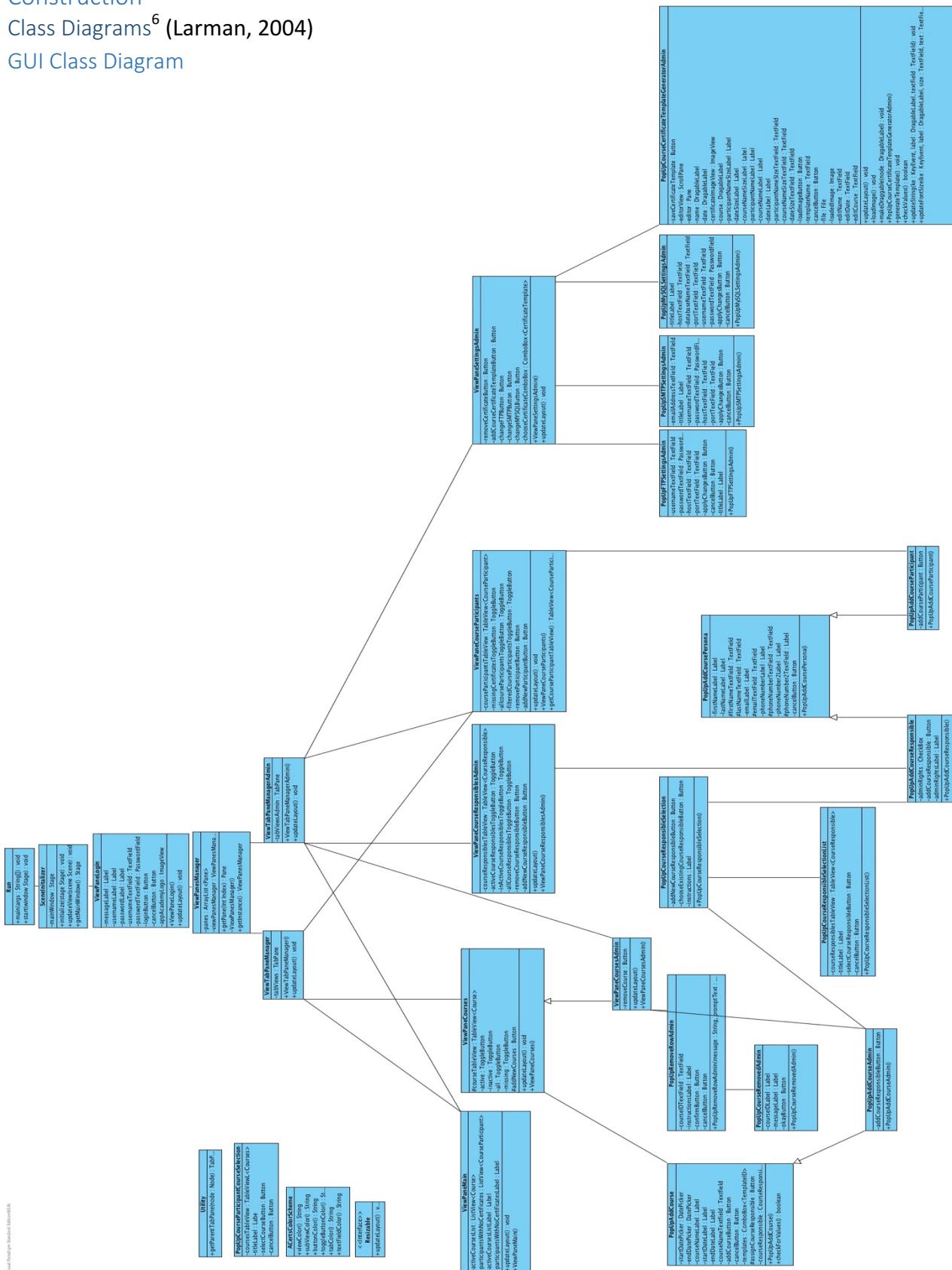


⁵ See Appendix 10 for High-Res Image

Construction

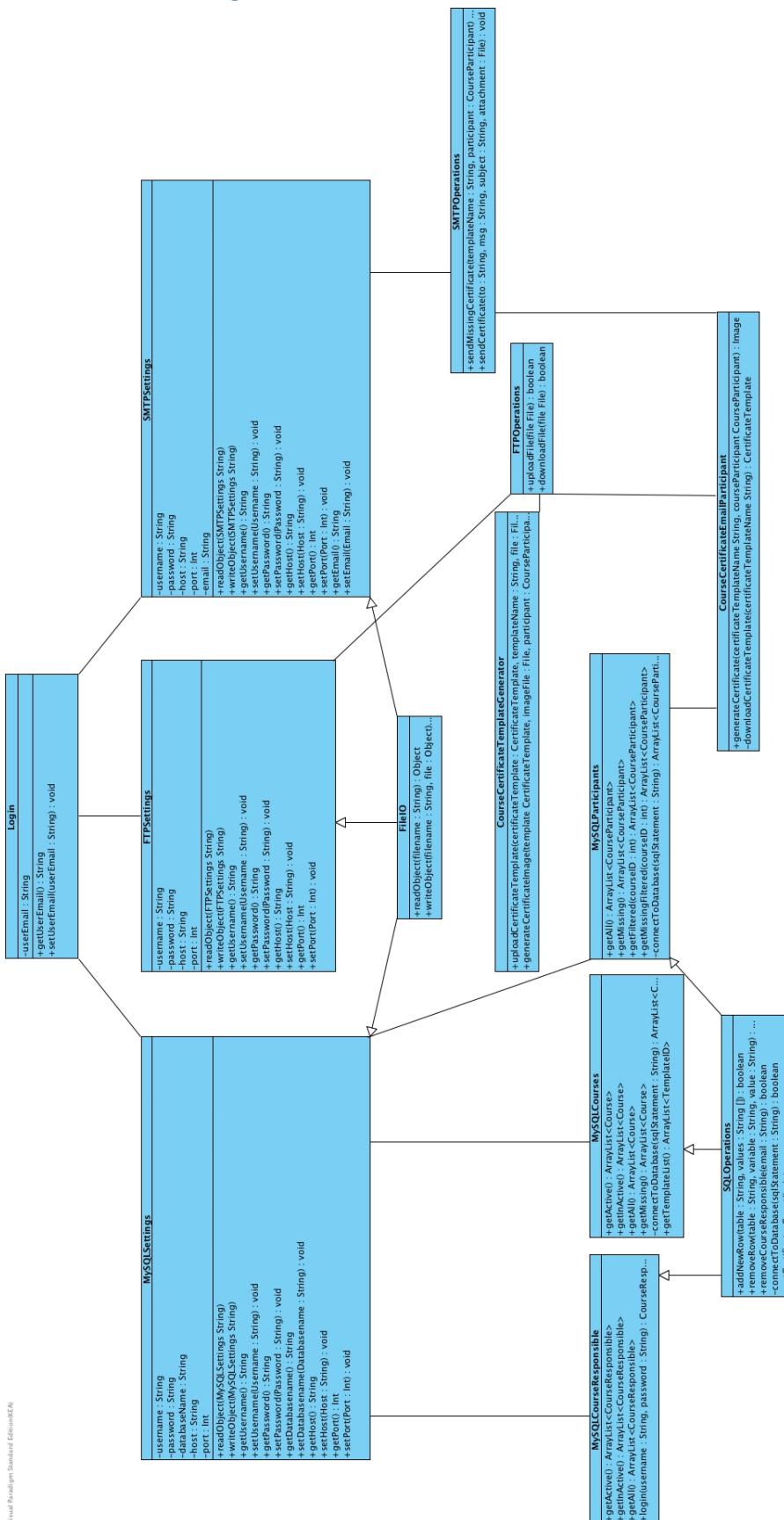
Class Diagrams⁶ (Larman, 2004)

GUI Class Diagram

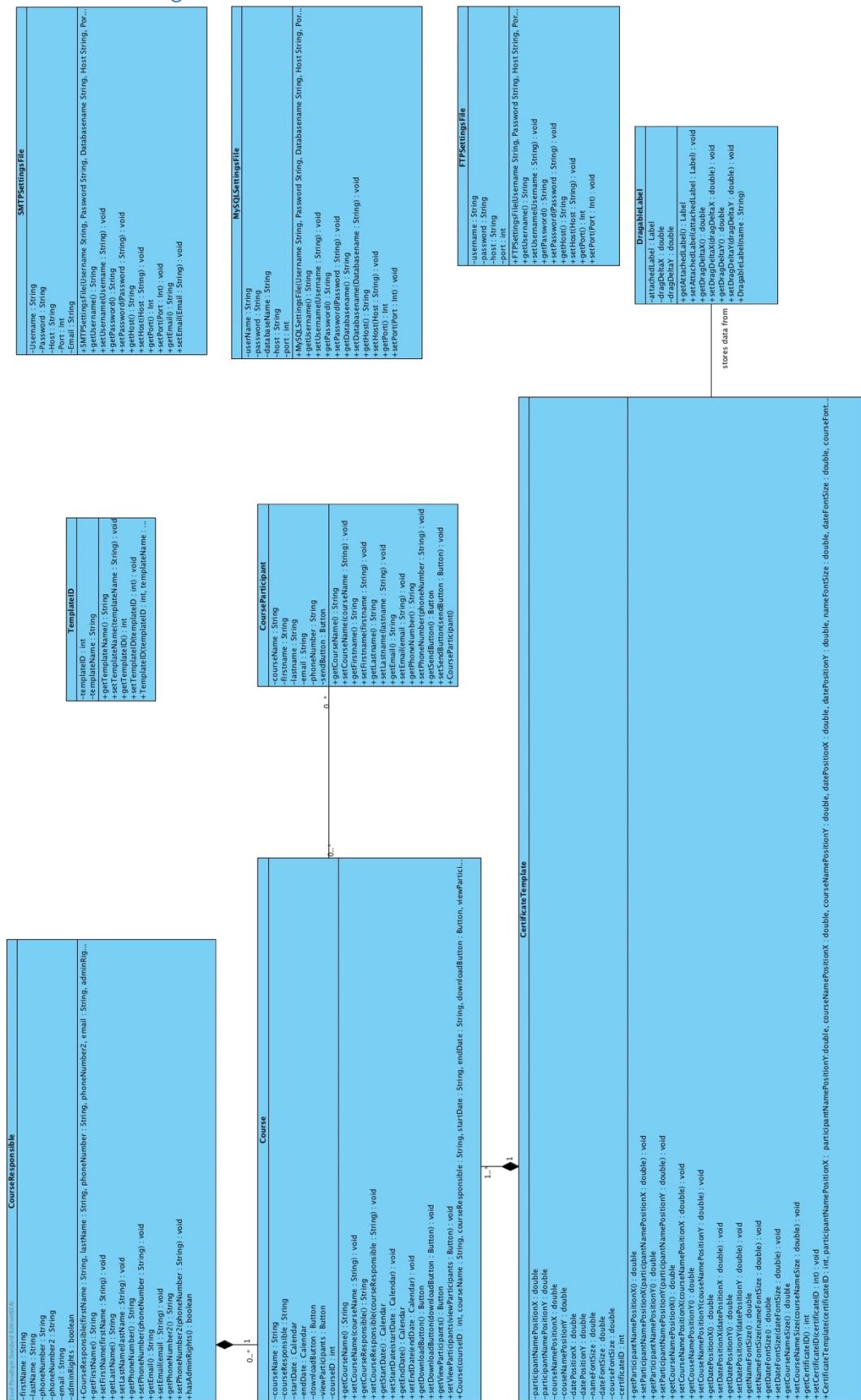


⁶ See Appendix 7 for High-Res Images

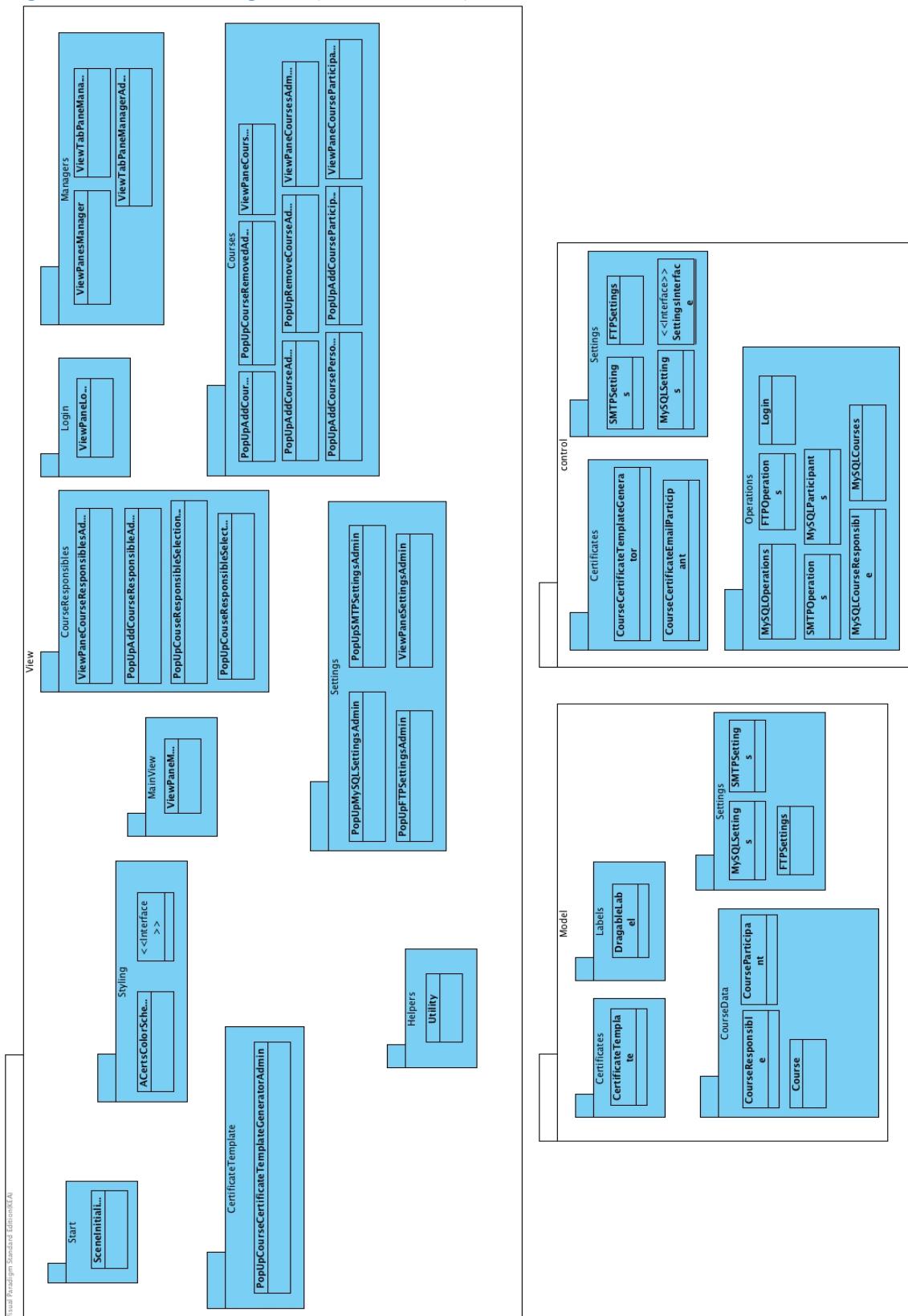
Functional Class Diagram



Model Class Diagram



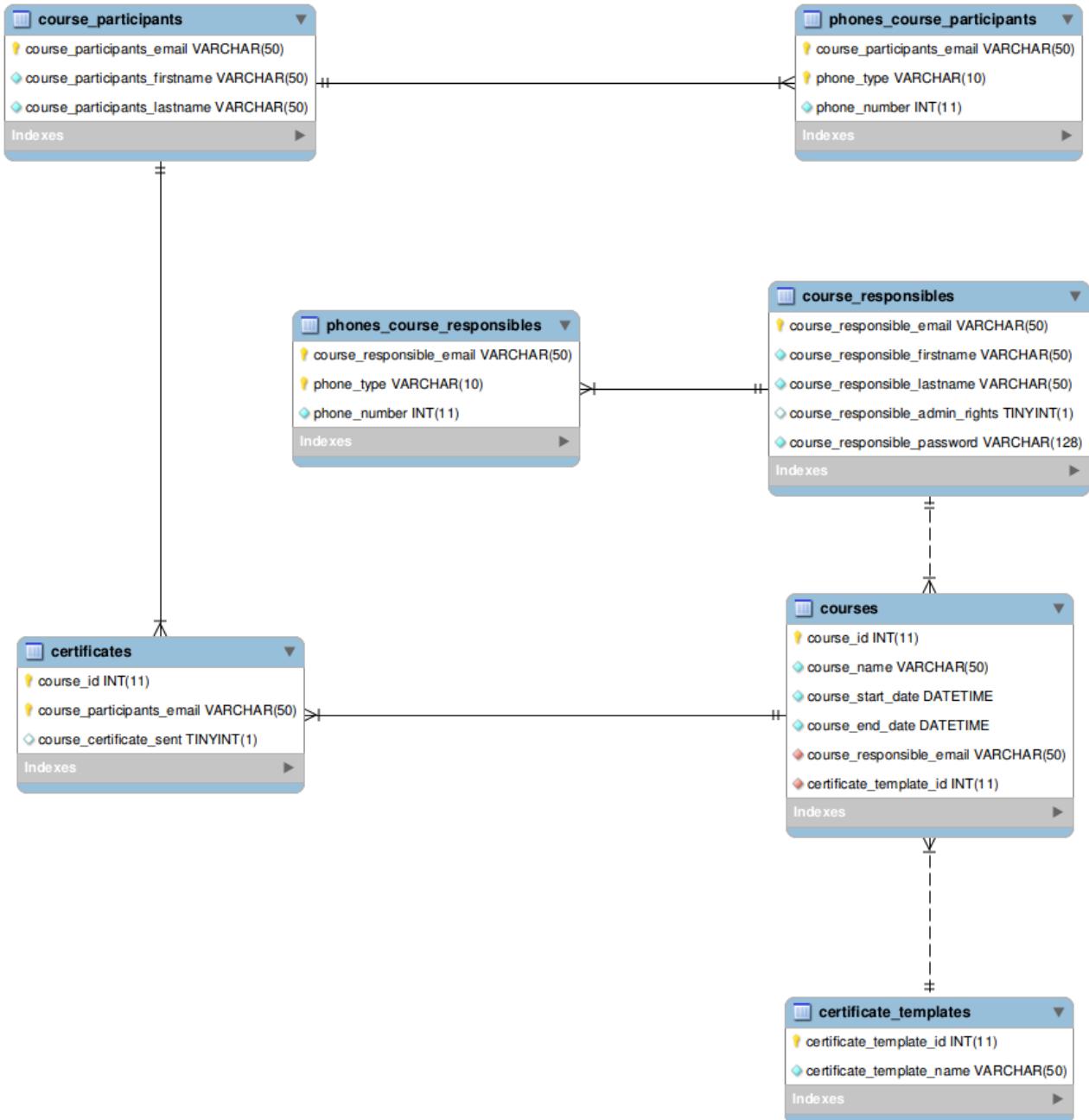
Logical Architecture Diagram⁷ (Larman, 2004)



⁷ See Appendix 8 for high resolution image

Database

Entity-Relationship Diagram



Database Normalization Process⁸ (Mr. B's Code Academy, 2012)

1st Normal Form – Verification Questions

A table is in first normal form if:	
1. There are no repeating groups:	Y
2. All data values are atomic:	Y
3. Each field has a unique name	Y
4. It has a primary key	Y

2nd Normal Form – Verification Questions

A table is in second normal form (2NF) if:	
1. It is first in First Normal Form (1NF)	Y
2. All non key attributes are dependent on ALL PARTS of the primary key	Y
3. Each field has a unique name	Y
4. It has a primary key	Y

3rd Normal Form – Verification Questions

A is in third normal form (3NF) if:	
1. It is first in Second Normal Form (2NF)	Y
2. All non key attributes are not dependent on any other non-key attribute	Y
3. Each field has a unique name	Y
4. It has a primary key	Y

⁸ See Appendix 9 for Original Excel File (For Better Visibility)

key	3NF
	Course Name
PK	CourseID
FK	Course Responsible Email
FK	Certificate Template ID
	Course Start Date
	Course End Date
PK	Course Responsible Email
	Course Responsible First Name
	Course Responsible Last Name
	Course Responsible Admin Rights
PK	Certificate Template ID
	Certificate Template Name
PK	Course Participant Email
	Course Participant First Name
	Course Participant Last Name
CPK	CourseID
CPK	Course Participant Email
	Course Participant Certificate Start
CPK	Course Responsible Email
CPK	Phone Type
	Course Responsible Phone Number
CPK	Course Participant Email
CPK	Phone Type
	Course Participant Phone Number

keys	1NF
	Course Name
PK	CourseID
FK	Course Responsible Email
FK	Certificate Template ID
	Course Start Date
	Course End Date
PK	Course Responsible Email
	Course Responsible First Name
	Course Responsible Last Name
	Course Responsible Admin Rights
PK	Certificate Template ID
	Certificate Template Name
PK	Course Participant Email
	Course Participant First Name
	Course Participant Last Name
CRK	CourseID
CRK	Course Participant Email
	Course Participant Certificate Set
CRK	Course Responsible Email
CRK	Phone Type
	Course Responsible Phone Number
CRK	Course Participant Email
CRK	Phone Type
	Course Participant Phone Number

Database Integrity

We have deliberately chosen to destroy database integrity when it comes to the removal of a Course Responsible from aCerts. The reason is, that we want to preserve the Course Information and this means when a Course Responsible is removed he is only removed from the Course Responsible Table and the Courses he had assigned remain intact. This means that the Course Responsible is locked out from the system and can no longer access it, but the essential Data remains available.

Location in Source Code:

Line 41-49 in control/operations/SQLOperations.java

```
public static boolean removeCourseResponsible(String email)
{
    String sqlStatement1 = "SET FOREIGN_KEY_CHECKS=0;";
    String sqlStatement2 = "DELETE FROM course_responsibles\n" +
        "WHERE course_responsible_email = '" +email+ "';";
    String sqlStatement3 = "SET FOREIGN_KEY_CHECKS=1;";

    return connectToDatabaseMultiple(sqlStatement1, sqlStatement2, sqlStatement3);
}
```

Transition

Reviews

Review by Kristian during 2nd Iteration

Kristian tried the Prototype and was very pleased with the vision portrayed through it. We are content to proceed at this point, since there was no criticism.

Review by class member Lucas, 4th Iteration

Lucas tried out our program in it's final stage. He was overall pleased, he objected against our decision to place the functionality to create a Course Certificate Template within the Settings pane. He deemed it illogical and informed as that he would have preferred to have seen it as a separate pane all together. We conceded his point as being valid, but due to resource management we deemed it a low priority implementation and decided against changing the current layout.

Review with Kristian for deployment, 4th Iteration

We had a skype meeting with Kristian for our final review. We went through several scenarios in aCerts, creating courses, adding Participants and Course Responsibles, and also sending certificates to his different company email's to test functionality. He was highly appreciative of our efforts and congratulated us on a job well done. He expressed satisfaction and was looking forward to deploying the program within AppAcademy.

Program User-Manual / AppAcademy Certificates Manager

First Time Setup

When running the program for the first time, the program will generate a MySQL Settings file. Since there is not an existing MySQL Settings file then the program will connect to a MySQL server at local host, named 'AppAcademy' with the password 12345678.

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

When logged in the first time, all FTP Settings and SMTP Settings are not set so a lot 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 local host, if they are to use the same server.

[Creating a certificate template](#)

When creating a certificate template, you should have an image ready where it is 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 markers around with the example text to customize the certificate.

When positioning 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 participant's tab, and press the send button.

Cancelled Features

['View as User X'](#)

We had a feature idea that can be seen in prototype during Iteration 1+2, where any Admin-level user would be able to change view to different Users to see their work. We decided against this feature later on as we deemed it an unnecessary implementation taking resources into consideration, that were better spent on improving other things.

['Password Reset'](#)

Early on we wanted a password Reset feature, but decided against this and instead made it so that Kristian would be able to hardcode it instead. A nice to have feature but considering the scope and user base for the tool, highly unnecessary.

['Download/Upload Course Material'](#)

This is a feature that we still would want in our program, we've had to cut it out due to time constraints. Sadly, this won't be in our deployed version. The idea was to have course material uploaded with courses so that one could always revise courses and have all information accessible through our aCerts tool.

['Statistics'](#)

Statistics, is something that we would like to have. Yet again we deemed it unnecessary considering AppAcademy's size and the scope of the project. Anyway Kristian would always be able to query the database himself since he's proficient in MySQL.

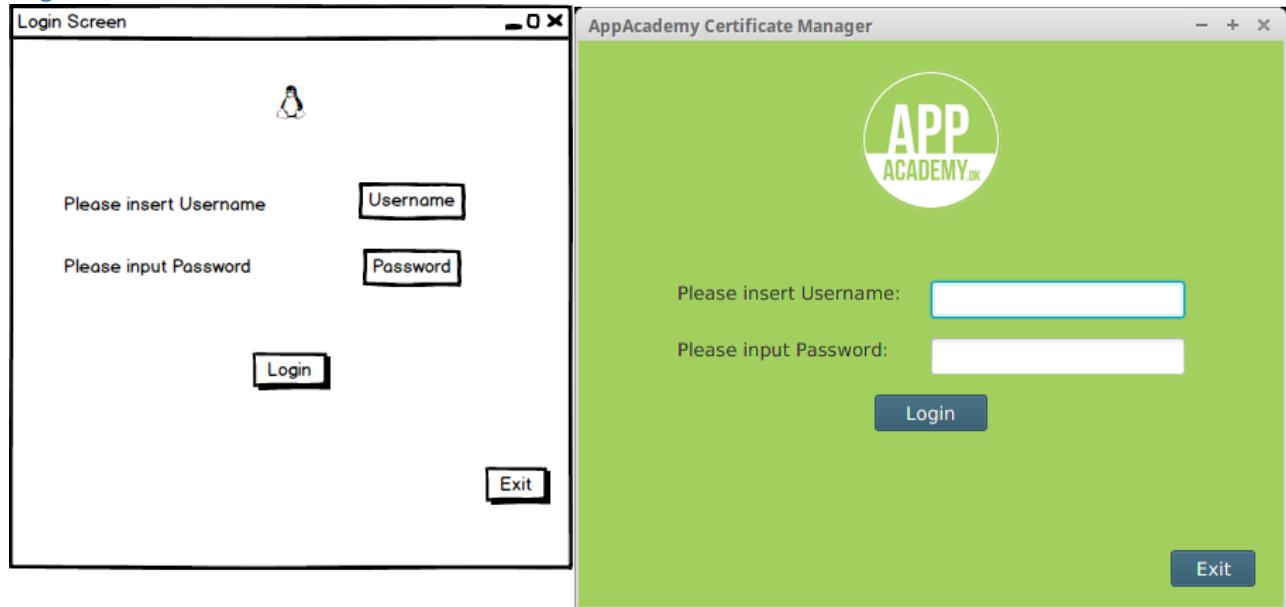
['Language File'](#)

A nice to have feature that simply didn't make the cut. It got cancelled during our 2nd Iteration, simply unnecessary considering the program is only used in Denmark and AppAcademy has no international offices. During our review with Kristian we discussed it, and he didn't consider it interesting in comparison with other features.

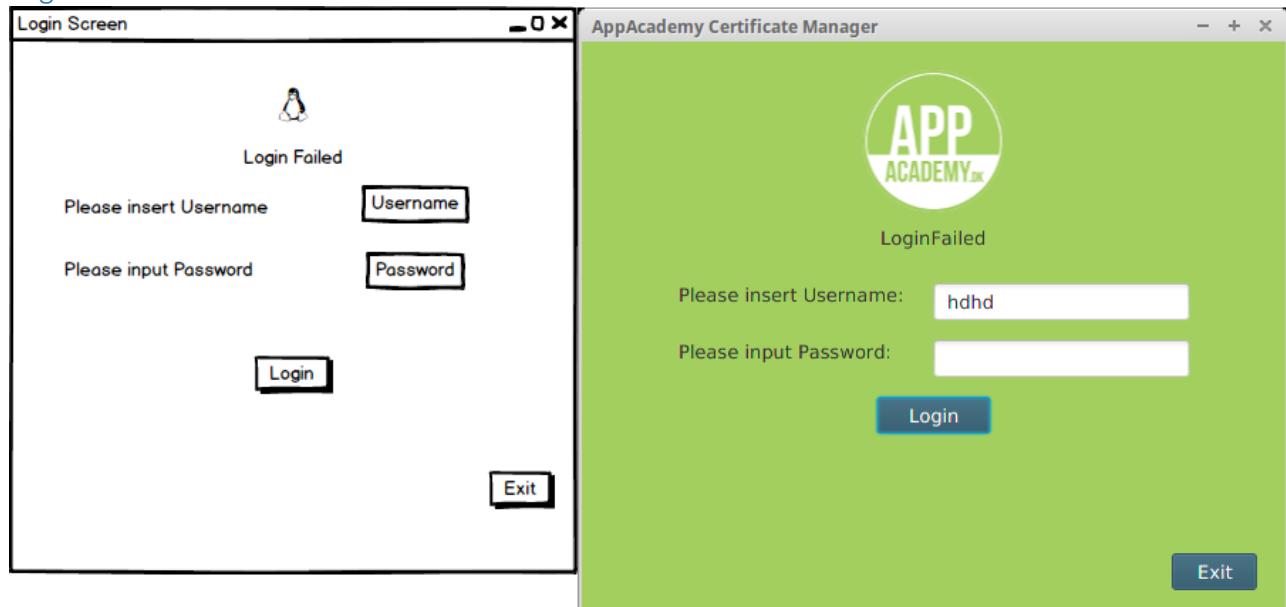
aCerts – Prototype contra actual Program⁹

We've fully maintained a functional Prototype during the course of development, we only stopped updating and maintaining it at the end of our 3rd Iteration at which point we had full focus on the actual aCerts program itself. The prototype can be tried using a program called [Balsamiq Mockups 3](#). The Prototype files are provided as Appendices, so that if the reader would want they can try the prototype throughout different iterations where changes have been vast.

Login Screen



Login Failed



⁹ See Appendix 6 for high resolution originals.

Main View

The 'Home Screen' window displays 'Active Courses' (Course 1, Course 2, Course 3) and 'Missing Certificates' (Course 1, Course 2, Course 3). The 'AppAcademy Certificate Manager' window shows 'Active Courses' (InfoSec, SWD, SWD, SWD) and a list of 'Missing certificates' (William Shakespeare, Ernest Hemingway, Tony TwoTimes, Johnny BeGood, Christopher Hitchens, John Dennet, Scott Fitzgerald, Opus Dei, Brun Sovs, Franklin Delano Roosevelt).

Courses View

The 'Courses View' window lists courses: Active (3D Printer, Business CRM Systems), InActive (Missing Certs), and All (Missing Certs). The 'AppAcademy Certificate Manager' window shows a table of courses with columns: Course, Course Responsible, Start Date, End Date, and Participants. It includes a 'Download' button and a 'View' button for each row.

Course Participants View

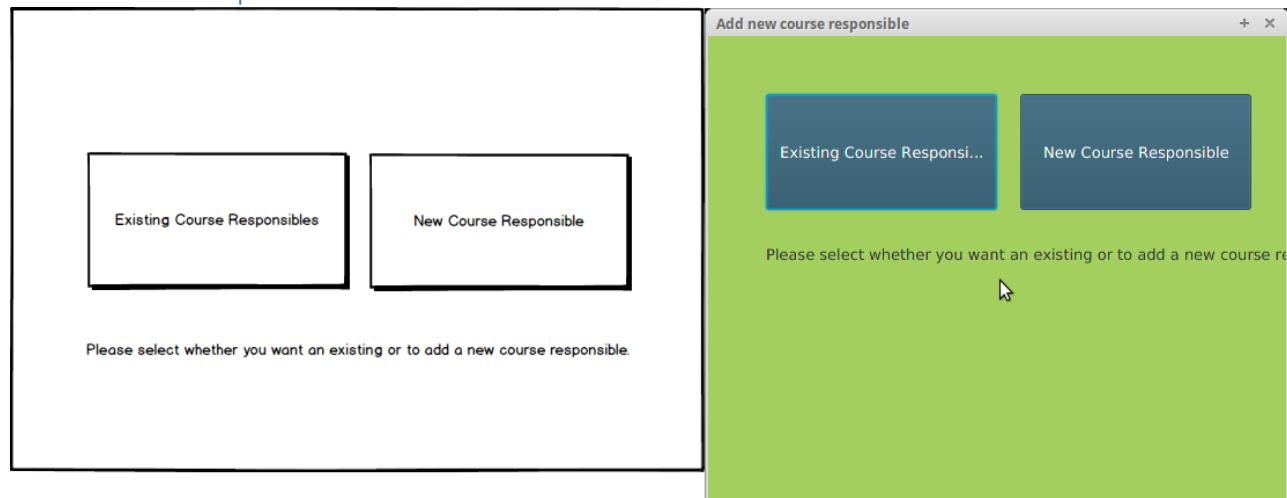
The 'Course Participants View' window lists participants: All (Max Maximus, Dorte Dorteimius Dorteimian), Missing Cert (Dorte Dorteimius Dorteimian), and Filtered (Dorte Dorteimius Dorteimian). The 'AppAcademy Certificate Manager' window shows a table of participants with columns: Course, First Name, Last Name, Email, Phone Number, Phone Number 2, certificate Sent, and Send Certificate. It includes a 'Send' button and a 'Send' link for each row.

Course Responsibles View

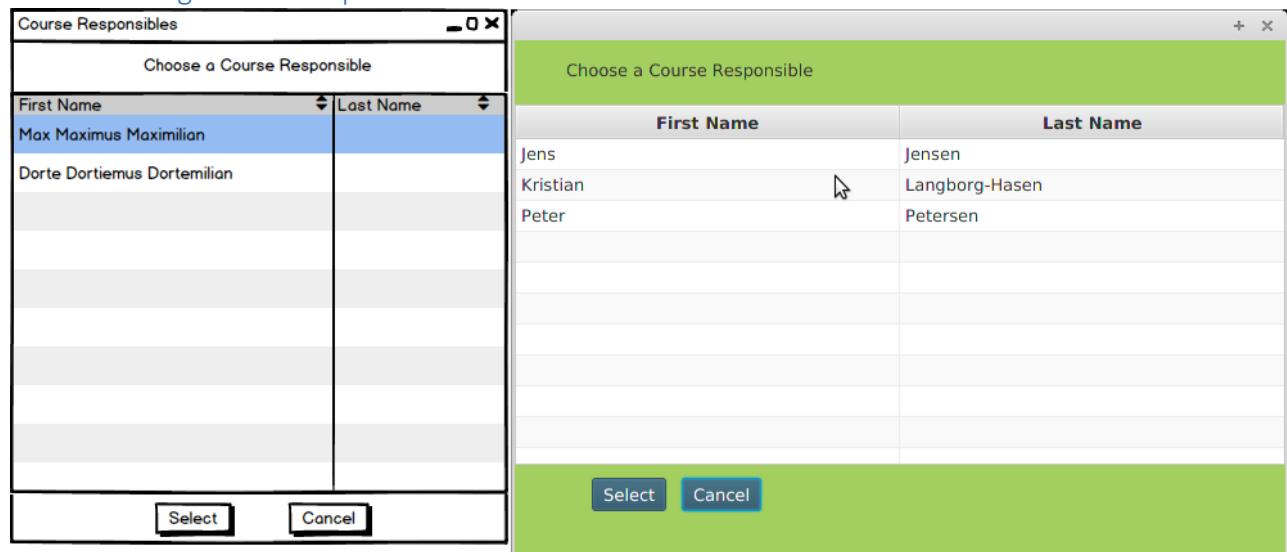
Settings View

Add New Course

Select Course Responsible for Course



Select Existing Course Responsible for Course



Remove Course

Please enter Course ID for the course you want to remove
This is Permanent

Course ID

Confirm Cancel

Remove Course

Warning!! This is permanent and cannot be undone!
Please input a course ID for the course
that you want to remove

Input course id...

Confirm Cancel

Course Removed

SYS Message

Course Name

Course ID

Course has now been removed.

Okay

1

Course successfully removed

Okay

Add New Course Participant

Add Course Participant

First Name E-mail

Last Name

Phone # Phone #2

Add Course Participant Cancel

Add new Course Participant

First name: E-mail:

Last name:

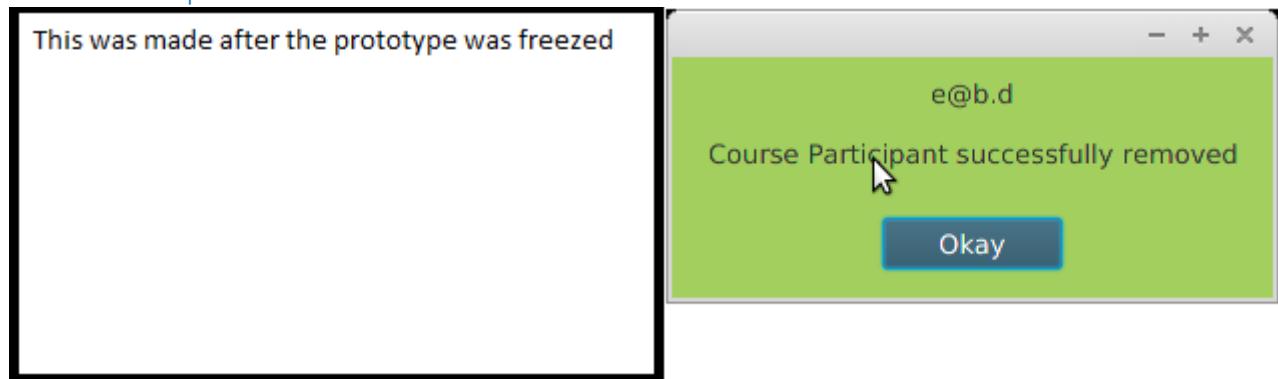
Phone # Phone #2

Add course participant Cancel

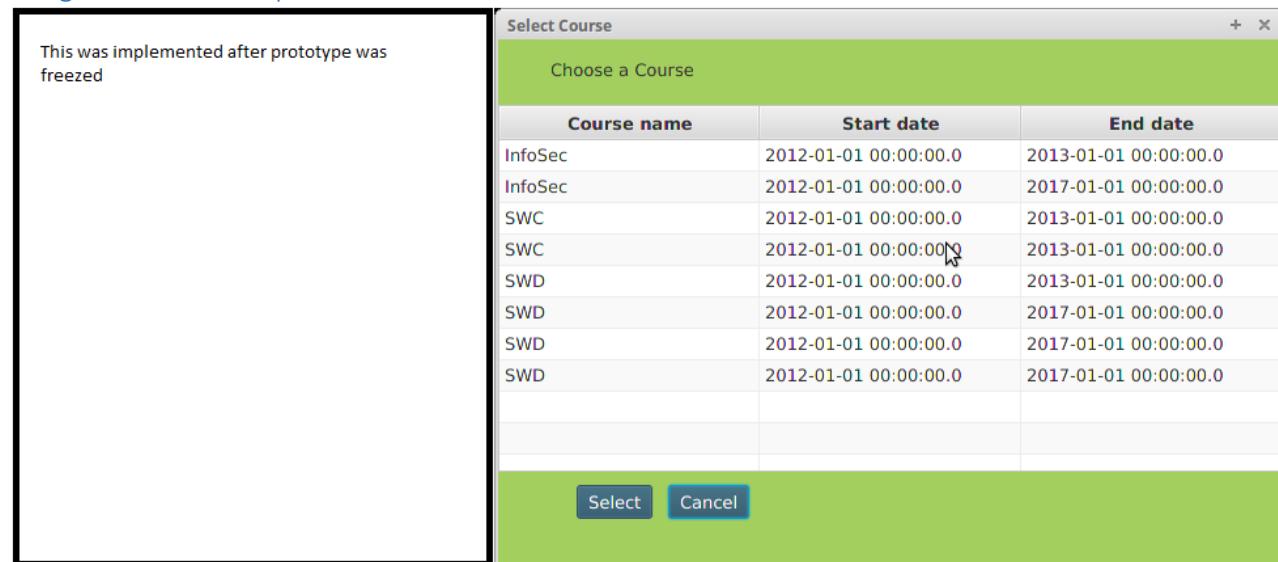
Remove Course Participant



Course Participant Removed



Assign Course Participant to Course



Add New Course Responsible

Add Course Responsible

First Name: E-mail:

Last Name: Choose Course

Admin Rights

Phone #: Phone #2:

Add new Course Responsible

First name: E-mail:

Last name:

Admin rights

Phone #: Phone #2:

Password:

Remove Course Responsible

This window was implemented after the prototype was freezed

Remove Course Responsible

Warning!! This is permanent and cannot be undone!
Please input the email address for the course responsible that you want to remove.

NOTICE: The course responsible will not be able to login afterwards!

Course Responsible Removed

This was implemented after the prototype was freezed

peter@petersen.dk

Course Responsible successfully removed

Change FTP Settings

FTP Settings

Username Password

Host Port

Apply Changes Cancel

FTP Settings

Username Password

Host Port

Apply cha... Cancel

Change MySQL Settings

MySQL Settings

Host

Database Name Port

Username Password

Apply Changes Cancel

MySQL Settings

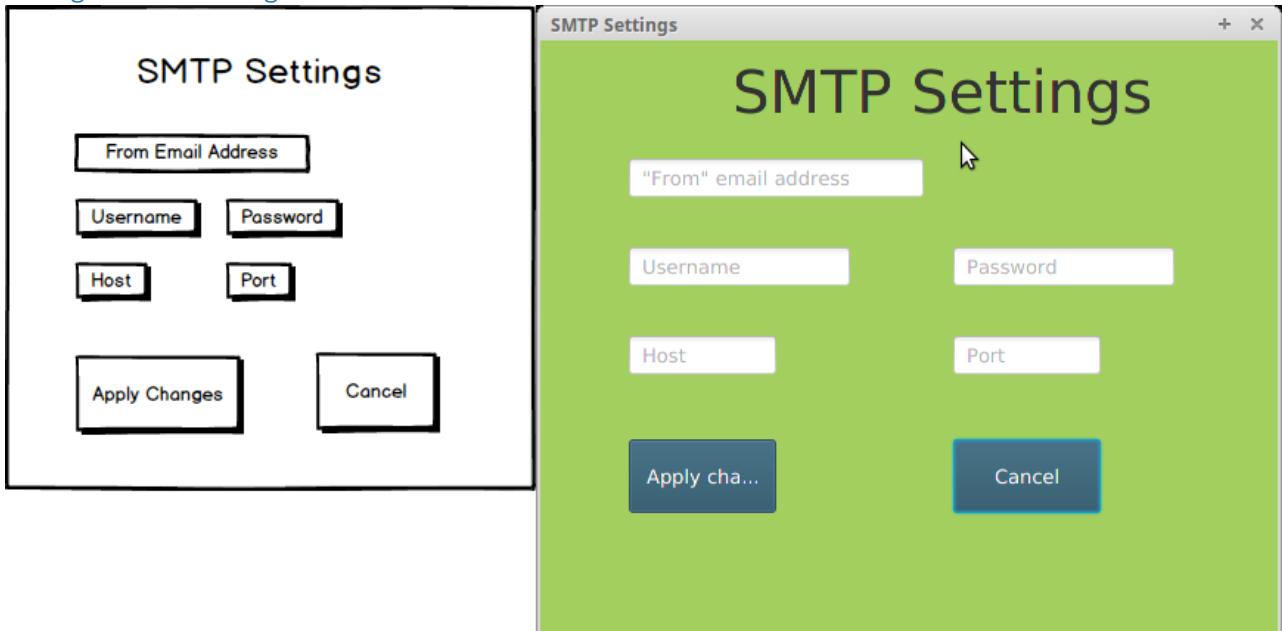
Host

Database Na Port

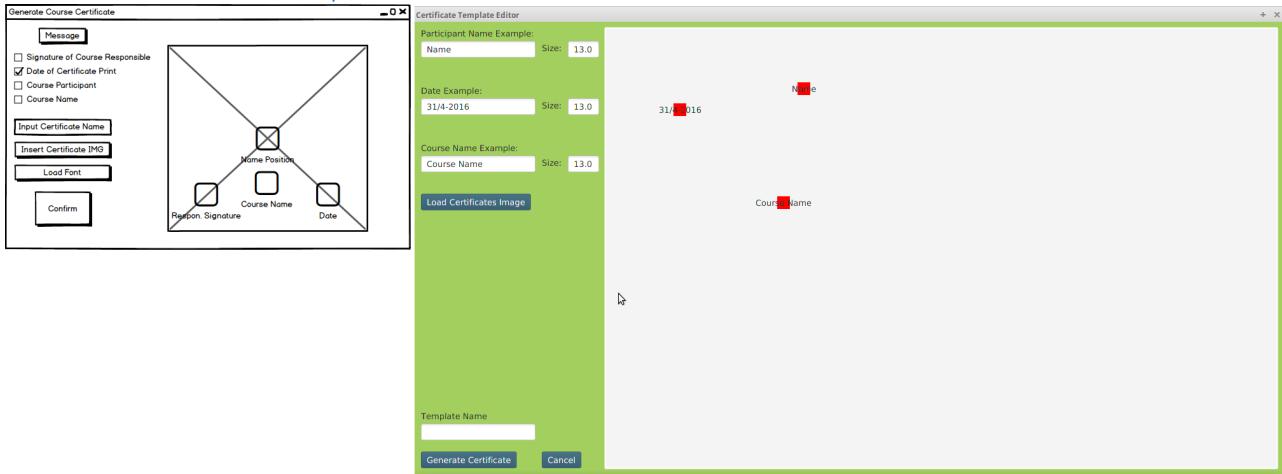
Username Password

Apply cha... Cancel

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 have 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 data source 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.

```

public synchronized void moveBall() {

    double minX, minY, maxX, maxY;

    minX = minY = 0.0;
    maxX = scene.getWidth() - radius;
    maxY = scene.getHeight() - radius;

    // Change the direction of the ball
    if (this.x < minX || this.x >= maxX)
        this.deltaX *= -1;
    if (this.y < minY || this.y >= maxY)
        this.deltaY *= -1;

    this.x += this.deltaX;
    this.y += this.deltaY;

    // Repaint the ball with the new coordinates
    this.ball.relocate(this.x, this.y);
}

```

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.

```

@Override
public void run() {
    while (true) {
        try {
            synchronized(scene) {
                this.moveBall();
            }
        }
    }
}

```

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

[Semaphore](#) (Tanenbaum & Bos, n.d.) (Silberschatz, 2013)

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

The standalone program is a jar file which contains all needed libraries.

Initially we weren't going to implement the "Drag 'n' Drop" functionality if we couldn't find the time and resources, but the customer thought it would make sense to implement it, so we reallocated time and resources and made it a priority feature. Unfortunately, we couldn't find time to implement upload and download of course material functionality, but the customer didn't find it essential, so the two functionalities in terms of resources kind of squared even. During our 2nd Iteration we found out that we forgot to implement the functionality to send certificates to participants which we had as a primary requirement. After this we ended up implementing it and learned that its possible to forget essential parts as you can be tricked by other requirements gathered later on. This took us very much by surprise, since we were being very thorough.

As can be seen in the captain's log we miscalculated our man hours but the buffer made up for this. Also we didn't use as much time as budgeted in the design part whereas the construction part used much more. These two parts balanced out our poor assessment and in the end, we've come out with a surplus of man hours on our total budget.

Bibliography

- Anon., n.d. *projekt_og_analyseredskaber.pdf (Figur 4.6)*. s.l.:s.n.
- Bruegge, B. & Dutoit, A. H., 2010. *Object-Oriented Software Engineering*. 3rd Edition ed. s.l.:Pearson.
- Ema, D., 2016. *Feasibility study.ppt*. s.l.:s.n.
- Kastner, R., 2011. *Why Projects Succeed - Proactive Risk Strategy*, s.l.: s.n.
- Larman, C., 2004. *Applying UML and Patterns, Chapter 6*. 3rd Edition ed. s.l.:s.n.
- Larman, C., 2004. *Applying UML and Patterns, Chapter 10*. 3rd Edition ed. s.l.:Addison Wesley Professional.
- Larman, C., 2004. *Applying UML and Patterns, Chapter 15*. 3rd Edition ed. s.l.:Addison Wesley Professional.
- Larman, C., 2004. *Applying UML and Patterns, Chapter 16*. 3rd Edition ed. s.l.:s.n.
- Larman, C., 2004. *Applying UML and Patterns, Chapter 34*. 3rd Edition ed. s.l.:s.n.
- Larman, C., 2004. *Applying UML and Patterns, Chapter 5, Section 4*. 3rd Edition ed. s.l.:Addison Wesley Professional.
- Larman, C., 2004. *Applying UML and Patterns, Chapter 6, Section 17*. 3rd Edition ed. s.l.:Addison Wesley Professional.
- Larman, C., 2004. *Applying UML and Patterns, Chapter 9*. 3rd Edition ed. s.l.:Addison Wesley Professional.
- Larman, C., n.d. *Applying UML and Patterns, Chapter 16+22*. 2nd Edition ed. s.l.:s.n.
- Larman, C., n.d. *Applying UML and Patterns, Chapter 23*. 2nd Edition ed. s.l.:s.n.
- Larman, C., n.d. *Applying UML and Patterns, Chapter 30*. 2nd Edition ed. s.l.:s.n.
- m.fl., H. J., 2012. *Organisation*. 5th ed. Copenhagen: Trojka / Gads.
- m.fl, H. J., 2012. *Organisation*. 5th ed. Copenhagen: Trojka / Gads Forlag A/S.
- Mr. B's Code Academy, 2012. *YouTube*. [Online]
- Available at:
https://www.youtube.com/watch?v=HHDH6N_qjm4&list=PLqGRWdRe4qUAUjNuiEyNYAWI1kuwd8yve&spfreload=10
[Accessed 2016].
- Oracle, n.d. *Java Docs*. [Online]
- Available at: <https://docs.oracle.com/javase/8/docs/technotes/guides/#javavm>
[Accessed 19 April 2016].
- Oracle, n.d. *Java Docs*. [Online]
- Available at: <https://docs.oracle.com/javase/8/docs/technotes/guides/#javavm>
[Accessed 19 April 2016].
- Silberschatz, A., 2013. *Operating System Concepts*. 9th Edition ed. s.l.:Wiley.
- Steve McConnell, H., n.d. s.l.:s.n.
- Tanenbaum, A. S. & Bos, H., n.d. *Modern Operation Systems 4th edition, 2.3.5 Semaphores*. 4th Edition ed. s.l.:s.n.
- TV2, n.d. *TV2 Nyheder*. [Online]
- Available at: <http://nyheder.tv2.dk/2014-12-19-tjek-her-laegger-du-dig-mere-syg-end-gennemsnittet>
[Accessed 19 April 2016].

Bibliography Code

Web sources also commented where used in Source Code

<http://stackoverflow.com/questions/19548363/image-saved-in-javafx-as-jpg-is-pink-toned>

<http://www.codejava.net/java-se/networking/ftp/java-ftp-file-upload-tutorial-and-example>

<http://www.codejava.net/java-se/networking/ftp/java-ftp-file-download-tutorial-and-example>

http://www.tutorialspoint.com/javamail_api/javamail_api_send_email_with_attachment.htm

<http://stackoverflow.com/posts/10689478/revisions>

<http://stackoverflow.com/questions/33186784/javafx-get-access-to-a-tab-or-tabpane-from-a-tabcontentnode>

Appendices

Appendix 1 – Technical Feasibility

Appendix 2 – Technical Feasibility

Appendix 3 – Schedule Feasibility

Appendix 4 – Schedule Feasibility

Appendix 5 – Domain Model

Appendix 6 – Prototype Comparisons to Program Comparisons

Appendix 7 – Class Diagrams

Appendix 8 – Logical Architecture Diagram

Appendix 9 – Database, Excel File

Appendix 10 – Generate Certificate Template