Report

# Table of Contents

Table of Contents 2

Introduction 2

Problem Definition 2

Company Description 3

UP – Unified Process 3

Inception 3

Organisation Type 3

SWOT 3

Business Case 3

Vision 3

Stakeholder Analysis 3

Feasibility Studies 7

Risk Assessment 10

Elaboration 11

FURPS+ 11

Use Cases 13

System Sequence Diagrams 25

Domain Model 32

Construction 34

Database 34

Transition 34

Summary 34

Bibliography 34

Appendices 34

# Introduction

## Problem Definition

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

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

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

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

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

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

## Company Description

# UP – Unified Process

## Inception

### Organisation Type

### SWOT

Svar skal vœre korte og koncise.

|  |  |
| --- | --- |
| Strengths | Weaknesses |
| Hvilke fordele har AppAcademy?  Hvor differentierer AppAcademy sig fra konkurrenter?  AppAcademys økonomi, konsulenter osv. | Hvad kunne AppAcademy gøre bedre?  Hvad skal AppAcademy undgå?  Hvilke faktorer udløser tab af afholdte kurser? |
| Opportunities | Threats |
| Hvilke muligheder kan AppAcademy se I fremtiden?  Hvilke trends gør sig gœldende indenfor markedet pt?  Hvilke teknologiske innovationer vil gavne AppAcademy? | Hvilke udfordringer står AppAcademy overfor?  Hvad gør AppAcademies konkurrenter der kunne indlfyde vores position?  Er den teknologiske udvikling med f.eks. e-kurser en trussel?  Er der kaptial problemer?  Er virksomheden sårbar pga. Størrelsen? |

### Business Case

### 

### Vision

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

### Stakeholder Analysis

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

PLACEHOLDER FOR FIG. 12.7 I ORGANISATION 5. UDGAVE

Grey Eminence: KEA

Hostage/ResourceStakeholder: Kristian

Hostage: Course Responsibles

ResourceStakeholder: Project Group

External Stakeholder: Testers

#### Grey Eminence: KEA

KEA as a Stakeholder:

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

KEA’s attitude towards the project:

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

KEA’s potential for conflict:

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

KEA’s resources:

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

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

App Academy as a Stakeholder:

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

AppAcademy’s attitude towards the project:

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

AppAcademy’s potential for conflict:

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

AppAcademy’s resources:

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

#### Hostage: Course Responsibles’

Course Responsibles’ as Stakeholders:

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

Course Responsibles’ attitude towards the project:

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

Course Responsibles’ potential for conflict:

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

Course Responsibles’ resources:

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

#### Resource Stakeholder: Project Group

Project Group as a Stakeholder:

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

Project Group’s attitude towards the project:

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

Project Group’s potential for conflict:

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

Project Group’s resources:

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

#### External Stakeholders: Testers

Testers as Stakeholders:

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

Testers’ attitude towards project:

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

Testers’ potential for conflict:

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

Testers’ resources:

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

### Feasibility Studies

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

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

Risk Assessment

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

#### Team Thoughts concerning Technical Feasibility

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

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

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

Dates where we will be working on the project:

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

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

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

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

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

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

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

Budgeted hours:

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

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

#### Team Thoughts regarding Schedule Feasibility

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

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

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

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

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

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

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

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

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

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

## Elaboration

### FURPS+

#### Functional

##### Features

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

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

##### Security

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

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

#### Usability

##### Human Factors

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

##### Documentation

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

A javadoc is also supplied with the source code.

##### Help

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

#### Reliability

##### Frequency of failure

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

##### Recoverability

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

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

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

##### Predictability

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

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

#### Performance

##### Response time

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

##### Resource usage

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

It will be scalable because there are no computations. For creating courses, participants and responsibles 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 optimised in the database.

#### Supportability

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

#### Legal(+)

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

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

### Use Cases

#### Fully Dressed - Generate Certificate Template

**Primary Actor:** Course Responsible

**Stakeholders and Interests:**

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

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

**Preconditions:**

a. Course is logged in and authenticated.

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

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

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

**Postconditions:**

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’.

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 Home Screen 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.

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

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

a. System crashes.

b. Admin corrects input and resumes.

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

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

11. System crashes during safe.

a. Admin verifies whether data was saved anyway.

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

**Special Requirements:**

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

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

-Admin has peripherals: Keyboard, Mouse, Monitor.

-Admin has access to database file.

-Admin has access to his OS file system.

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

**Open Issues:**

-File type variations.

-Font type variations.

-OS GUI variations (example. Arch Linux).

#### Fully Dressed - Generate Certificate Template

**Primary Actor:** Course Responsible

**Stakeholders and Interests:**

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

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

**Preconditions:**

a. Course is logged in and authenticated.

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

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

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

**Postconditions:**

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’.

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 Home Screen 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 realises 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 recognise file.

a. Admin tries again, system acknowledges IMG file.

b. Admin fails to upload an accepted format.

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

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

a. System crashes.

b. Admin corrects input and resumes.

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

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

11. System crashes during safe.

a. Admin verifies wether 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 filesystem.

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

**Open Issues:**

-File type variations.

-Font type variations.

-OS GUI variations(example. Arch Linux).

#### Fully Dressed: Send Missing Certificate

**Primary Actor:** Course Responsible

**Stakeholders and interests:**

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

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

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

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

**Preconditions:**

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

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

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

**Postconditions:**

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

**Main Success Scenario:**

1. Course Responsible logs into system.

2. Course Responsible selects "Courses" pane.

3. Course Responsible clicks 'List Courses'.

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

5. Course Responsible presses 'View Participants' at the Course he wants to send certificates for.

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

7. Course Responsible presses ‘View missing Certificate’ and is presented with a filtered List of Course Participants that havn’t received their certificate.

8. Course Responsible presses ‘Send Certificate’ for desired Course Participants.

9. Course Responsible is prompted with a confirmation message of sending.

10. 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 list courses.

a. The course responsible attempts a system restart.

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

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

a. Course responsibles 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 responsibles 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 filesystem.

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

#### Fully Dressed: Add Course

**Primary Actor:** Course Responsible

**Stakeholders and Interests:**

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

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

**Preconditions:**

a. Course Responsible is logged in and authenticated.

b. Systems are operational.

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

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

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

**Main Success Scenario:**

1. Course Responsible arrives at Home Screen.

2. Course Responsible selects ‘Courses’ pane.

3. Course Responsible clicks ‘Add Course’ 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 presented with ‘Course List’ view in database.

**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 course” 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 Course” button should now work as intended.

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

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

b. The course can now be created.

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

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

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

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

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

a. See step 1. a.

b. Verify the course has been added correctly.

**Special Requirements:**

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

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

-Course Responsible has access to internet/intranet.

-Course Responsible has access to his OS file system.

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

**Open Issues:**

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

#### Casual: Remove Course

**Primary Actor:** Course Responsible

**Stakeholders and Interests:**

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

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

**Preconditions:**

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

**Main Success Scenario:**

1. Course Responsible successfully authenticates.

2. Course Responsible proceeds to 'Settings' pane.

3. Course Responsible clicks 'Remove Course' button.

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

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

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

**Alternate Success Scenario:**

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

a. He retries with correct CourseID.

#### Casual: Remove Course Participant

**Primary Actor:** Course Responsible

**Stakeholders and interests:**

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

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

**Preconditions:**

Course responsible has successfully authenticated and landed on home screen.

**Main Success Scenario:**

1. Course Responsible opens the “Courses” pane.

2. Course Responsible clicks the “List courses” button.

3. Course Responsible clicks the “View” button besides the desired course.

4. Course Responsible clicks on the course participant he/she wish to remove.

5. Course Responsible clicks the “Remove” button.

6. The the desired course participant is now removed.

**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 “List Courses” button does not work

a. The course responsible restarts the system.

3. The course responsible selects the wrong course

a. The course responsible returns to the previous window with the table view and finds the right course.

#### Casual: Change FTP Settings

**Primary Actor:** Course Responsible

**Stakeholders and Interests:**

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

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

**Main Success Scenario:**

1. Course Responsible successfully logs in.

2. Course Responsible selects 'Settings' pane.

3. Course Responsible clicks 'Change FTP' button.

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

a. Username

b. Password

c. Host

d. Port

He fills in correct information.

5. Course Responsible applies new information to the system.

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

**Alternate Success Scenario:**

3. Course responsible doesn't have correct information.

a. He will contact App Academy for correct information.

#### Casual: Change SMTP Settings

**Primary Actor:** Course responsible

**Stakeholders and Interests:**

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

**App Academy:** Wants to the SMTP settings to be correct so the course responsibles 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.

b. He will contact App Academy for correct information.

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

Course responsible logs into the system and arrives at home screen. He/she opens the “Courses” pane and clicks the “List courses” button. Course responsible then clicks the “View” button besides the course where he/she wants to add a participant. Course responsible then clicks the “Add New” button. A window appears where the participant information has to be added which the course responsible does. The course responsible then clicks “Add Participant”.

#### Brief: Add Course responsible

Course Responsible with Admin Rights logs into the system. He proceeds to the Course Responsibles pane and presses Add Course Responsible. 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 wether or not this Course Responsible should have admin rights he can press Add Course Responsible. The Course Responsible has now been added and he is moved back to Course Responsible Tableview where the recently added should be present.Brief:

#### Remove Course Responsible

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

#### Brief: Remove Certificate Template

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

#### Brief: List Courses

Course Responsible logs into the system and arrives at home screen. Course Responsible opens the “Courses” pane which leads to a window where a “List courses” button appears. The button leads to a table view of the courses.

#### Brief: List Course Responsibles

Course Responsible with Admin Rights logs into the system. He proceeds to the Course Responsibles pane, here he presses the List Course Responsibles button. He is now presented with a list of all Course Responsibles that have active courses.

#### Brief: View Missing Certificate

Course Responsible logs into system and arrives at home screen. Course Responsible selects the "Courses" pane which leads to a window where a “List Courses” button appears. Course responsible clicks the button and a tableview of courses appears with a column consisting of “view” buttons. The Course Responsible can now click the view button for the desired course which will lead to a tableview which opens with a filter that shows the course participants who are missing course certificates.

#### Brief: View Active Courses

Course Responsible with no admin rights logs into the system. From here he proceeds to the Courses pane. Here is presented with a table view of his courses. This table is set to present Active Courses automatically.

#### Brief: View all Courses

Course Responsible with no admin rights logs into the system. From here he proceeds to the Courses pane. Here is presented with a table view of his courses. This table is set to present Active Courses automatically so he has to press 'All Courses' to be presented with a view of all courses at AppAcademy.

#### Brief: View inactive Courses

Course Responsible with no admin rights logs into the system. From here 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 'InActive Courses' in the upper left corner where the table will be filtered to show Courses that are from previous date than todays' date.

### Inception/Diagrams%201st%20Iteration/Generate%20Certificate%20Template.jpgSystem Sequence Diagrams

Figure . Generate Certificate Template

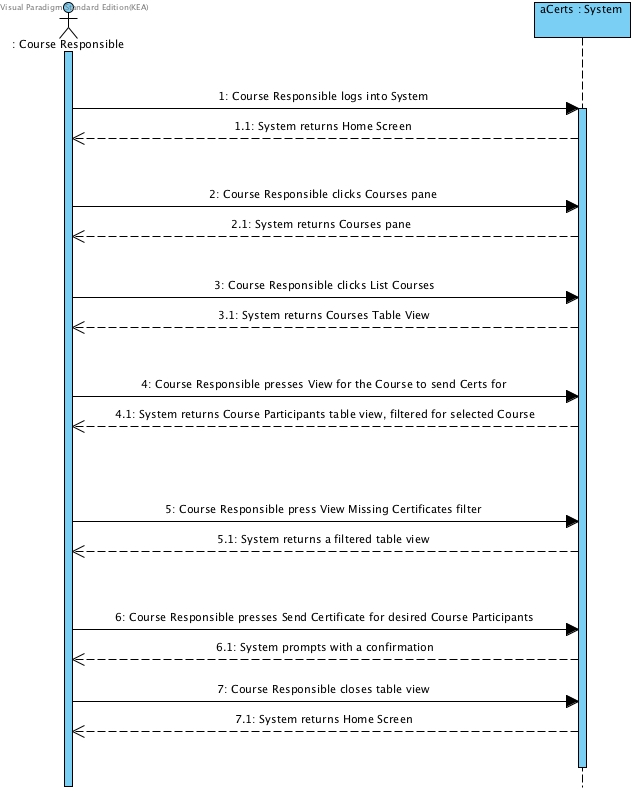


Figure . Send Missing Certificate

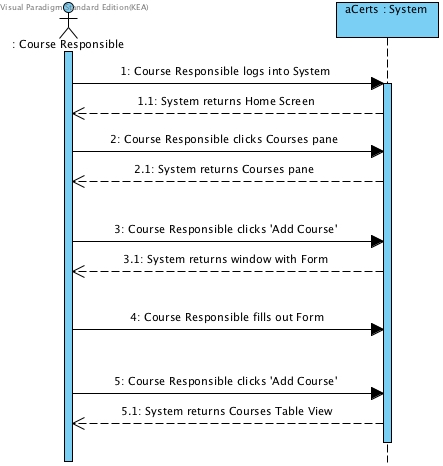


Figure . Add Course

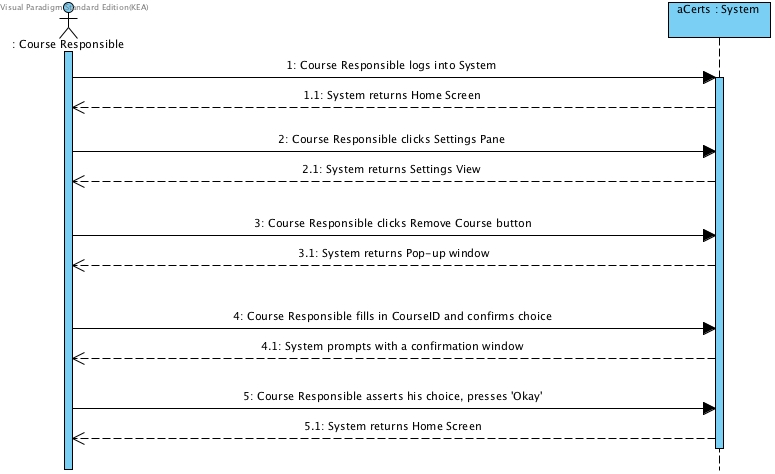


Figure . Remove Course

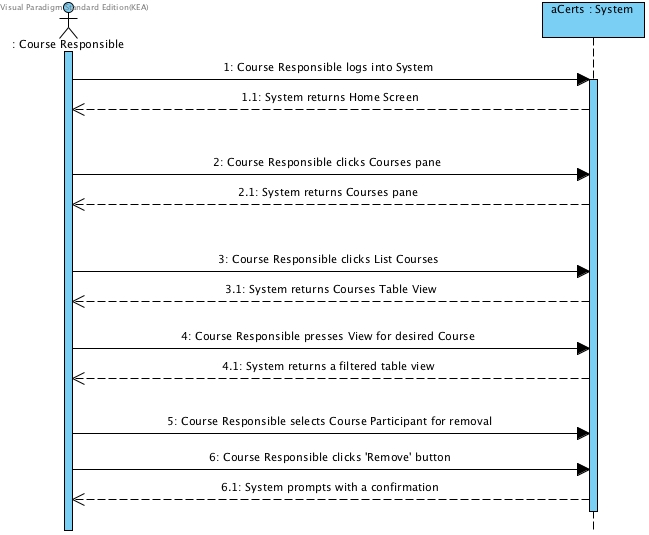


Figure Remove Course Participant

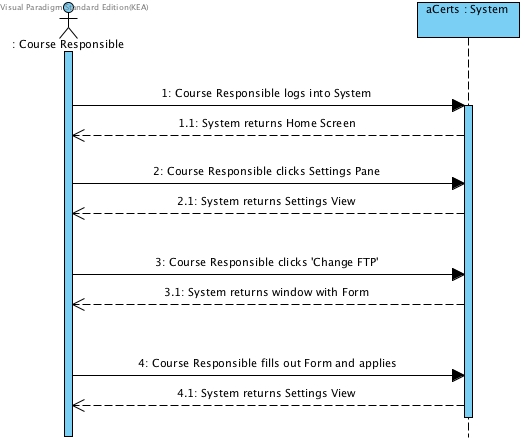


Figure . Change FTP Settings

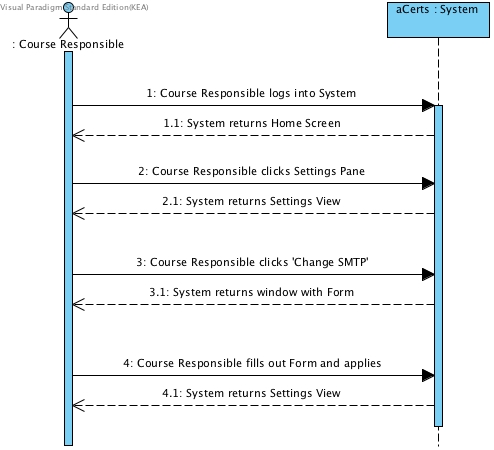


Figure . Change SMTP Settings

### Domain Model[[10]](#footnote-10)Elaboration/Domain%20Model.jpg

## Construction

### Database

## Transition

# Summary

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

# Bibliography

# Appendices

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