Logbook - BDSA15

Dennis Thinh Tan Nguyen, William Diedrichsen Marstrand, Jacob Mullit Mniche,
Thor Valentin Aakjr Olesen Nielsen

December 3, 2015

Contents

1 Log for date: 11/11-2015

Start: 14:00 — End: 16:00

Member attendence:

- Dennis Thinh Tan Nguyen attended.
- Jacob Mullit Mniche attended.
- Thor Valentin Aakjr Olesen Nielsen attended.
- William Diedrichsen Marstrand attended.

Meeting pins

- Set up Trello and Pomello
- Set up Log book
- Initialized C# program structure

Sprint Planning

- Sudy ConfigurationUI (Web Service Interface)
- Internal communication among subsystems
- Subsystems Classes
- Storage Interface
- Work Load
- Prviding Blue Team API
- Dependency Injections
- Using Adapter Pattern

2 Log for date: 11/12-2015

Start: 13:00 — End: 16:00

Member attendence:

- Dennis Thinh Tan Nguyen attended.
- Jacob Mullit Mniche attended.
- Thor Valentin Aakjr Olesen Nielsen attended.
- William Diedrichsen Marstrand attended.

Meeting pins:

- WebAPI (Fra Steven) - Adapter Pattern, Facade (Autosys Server)
- Initialized and refactored project structure - packages to projects
- UML Subsystem refactored
- UML Entity Object refactored
- Entities first

Branching Rules

- First create a local branch
- Then publish the branch so it resides on oring/master
- Required start name for naming convention for branches: implementation/, testing/, documentation/, bugfix/, refactor/

3 Log for date: 14/11-2015

Start: 12:30 — End: 15:10

Member attendence:

- Dennis Thinh Tan Nguyen attended.
- Jacob Mullit Mniche attended.
- Thor Valentin Aakjr Olesen Nielsen not attended.
- William Diedrichsen Marstrand attended.

Meeting pins:

- Creating overview of every subsystems
- Updated LucidChart of subsystems
- Defining classes and their responsibilites of every subsystems (Application logic and Storage)

Sprint Planning:

- Implement design decision for the new codeskeleton

4 Log for date: 11/16-2015

Start: 16 — End: 21

Member attendence:

- Dennis Thinh Tan Nguyen attended.
- Jacob Mullit Mniche not attended.
- Thor Valentin Aakjr Olesen Nielsen attended.
- William Diedrichsen Marstrand attended.

Meeting pins:

- Code Skeleton completed.
- Pull requests reviewed.
- Merged branches into master.
- Discussed about work processes and how we contact each other outside ITU. Overall conflict is that study related activities seem not to be separated from other activities thus causing a tense stressful environment for certain members. One proposal is to keep track of each members time board so that people know when they can contact each other or when they should not be disturbed. This will be done with a document showing everyones schedule and using status tags on Trello.

Sprint Planning:

- Need to plan future work and time for meetings to achieve a clear distinction between activities related to ITU and others.
-

5 Log for date: 11/17-2015

Start: 9:00 — End: 12:00

Member attendence:

- Dennis Thinh Tan Nguyen attended.
- Jacob Mullit Mniche not attended.
- Thor Valentin Aakjr Olesen Nielsen attended.
- William Diedrichsen Marstrand attended.

Meeting pins:

- Cleaned up code skeleton
- Added missing classes
- Implemented WebApi

Sprint Planning:

- Update SDD and RAD (UML subsystems, datafield definition, user definition, update entity object model)
- Create time schedule for all group members and dertermine meeting hours

Work planning:

- A Google Document has been made where the group members have written the time intervals where they are okay being contacted.
- In the document mentioned above working schemes have been made as well, indicating when group meetings are held.
- This is to separate study related activities from social activities thus avoid a stressful environment

6 Log for date: 11/24-2015

Start: 9:00 — End: 16:00

Member attendence:

- Dennis Thinh Tan Nguyen attended.
- Jacob Mullit Mniche attended.
- Thor Valentin Aakjr Olesen Nielsen attended.
- William Diedrichsen Marstrand attended.

Meeting pins:

- Implement code skeleton feedback
- Connect implementation tasks with functional requirements
- Fix UWP vs standard application issue

Sprint Planning:

- Need to fix subsystem dependencies with TAs by possibly using
- Connect functional requirements with implementation tasks
- Plan coding tasks for the next 4 weeks

Implementation priorities based on functional requirements subsystem dependencies:

Application logic

- UserManagement will be implemented first because WebApi and StudyManagement depend on it. We need to define how a User is to be represented so that SM can use WA.
- StudyManagement should be implemented secondly because its criteria and phase classes are used to define a study configuration. ProtocolManagement depends on a finalized study configuration.
- ExportManagement and PaperManagement can be implemented independent from rest of project.
- All subsystems depends on Repository with predefined CRUD operations.
- WebAPI depends on all subsystems based on their methods. We need to define what methods each package handler holds in order to develop our WebApi. The WebApi will be based on the deliveries from Steven.

Interface

- StudyConfigurationUI is independent from all subsystems in application logic. It is based on a web api that uses the server in the application logic. Should be able to retrieve teams stored in database and create a study define passed to the application logic.

User Validation The application logic will not handle user validation (e.g. manager or researcher), which is handled in the interface part of the system between yellow and blue part. The only thing application logic considers is whether a given user exists in the database. Roles are checked in Study Management based on an enum flagtype related to a user, e.g. a user can have a "Reviewer", "Manager" or "Validator" role determining which tasks are to be returned. A Reviewer will receive review tasks and a Validator will receive conflict tasks. Blue team will validate users based on whether they represent a Manager or Researcher.

Code Skeleton Design Changed to reduce coupling

If all packages have a facade interface used by other subsystems it allows single responsibility. Also allows team members to work on subsystems independently. We do not touch each other's code this way. A sub system should be able to be pulled out and replaced without causing trouble. By way of example, ExportManagement depends on the ProtocolManagement and its a predefined protocol. A using statement for Protocol is used in ExportManagement. Thus, we need to change the UML and use a dashed line between ExportManagement and ProtocolManagement. The dependency resides in where the object comes from and not how it is passed (e.g. from WebAPI).

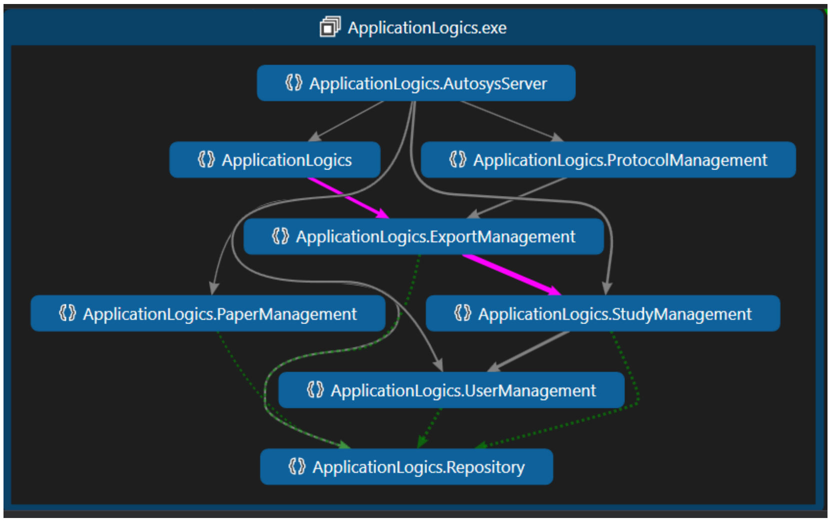
ApplicationLogics.exe
ApplicationLogics.AutosysServer
ApplicationLogics
ApplicationLogics.ProtocolManagement
ApplicationLogics.ExportManagement
ApplicationLogics.PaperManagement
ApplicationLogics.StudyManagement
ApplicationLogics.UserManagement
ApplicationLogics.Repository

Figure 1: Subsystem Dependencies

7 Log for date: 11/26-2015

Start: 9:00 — End: 14:00

Member attendence:

- Dennis Thinh Tan Nguyen attended.
- Jacob Mullit Mniche not attended.
- Thor Valentin Aakjr Olesen Nielsen attended.
- William Diedrichsen Marstrand attended.

Meeting pins:

- Rework database design
- Working on Criteria
- paperManagement

Sprint Planning:

- Talk with TA about database design
- Discuss Test driven development

Work planning:

- Add new title in each logbook, which detials any radical design changes.
- Thor and dennis has been working on a revicited entity relation model to be able to create the database
- We have desided to use a microsoft database to avoid any conflict between different types of databases
- We don't need to worry about database queing. Aparently c # will automaticly handled that aspect. According to Mokkel TA , each task will automaticly queue up to be processed. I don't think this will solve it on the client side, but that is hardly our problem
- You might want facilitate regular expressions in the search to enable more advanced searches. Instead of asking does this name appear in the text, then you can ask, I want to sample papers where this name appeared at least 8 times.

Work accomplished today:

- E
  R model created

- Subsytem dependency resolved: Create exportable item Interface To expose methods for formating exporting Criterias, Protocol and Criteria will implement CRUD
- Database Dispose have been replaced by Using blocks
- Database connection is maintained upon methods calls
- Created Facade package in application logic, used to translate entities to storage
- Repository have been moved to storage, and will now contain concrete repsoitories for all entities
- **William** is working on the Paper management class
- **Jacob** is working on the Phase and Criteria classes
- **Dennis** User management class
- **Thor** Storage, IFasade

8 Log for date: 12/01-2015

Start: 9:00 — End: 16:00

Member attendence:

- Dennis Thinh Tan Nguyen attended.
- Jacob Mullit Mniche attended.
- Thor Valentin Aakjr Olesen Nielsen attended.
- William Diedrichsen Marstrand attended.

Meeting pins:

- Testing strategy
- Shared data model required for use in repositories, stored model entities and testing
- Agreed on test syntax methodToTest_condition_expectedbehavior
- AutoMapper framework used to translate objects from application logic to DTO objects and Stored model entities
- Can now connect to a local database created from code first (AutoSysDbModel). Requires installation of SQL Server 2014 and login with (localdb)/myssql
- Agreed to prepare questions and think about potential conflicts involving work of others when implementing things. We now put more focus on preparations before group work. Members are expected to prepare questions and identify potential challenges involving the work of others.
- Reimplemented ids for entities in application logic where needed.

Sprint Planning:

- Sections from Testing Strategy document has been assigned to individual group members. Expected template delivery on Thursday the 3rd of December.
- Documentation boards on Trello are to be complete in order to clean up Trello for new tasks.

Current work:

- Thor is currently working on implementing the Storage repositories containing CRUD operations for all model entities. He will first implement the tests for each repository. This is now possible due to the agreed data model (had conflict with Paper and Criteria entity).

- Jacob is currently writing blackbox tests for the CriteriaHandler, working on a simple sorting operations which let a user sort Articles based on criterias and is working on Phase
- William is curently working on Papermanagement and exploring the nUnit framework (Which help reducing redundant test cases).
- Dennis working on black box testing and Fasades

9 Log for date: 12/03-2015

Start: 9:00 — End: 16:00

Member attendence:

- Dennis Thinh Tan Nguyen attended.
- Jacob Mullit Mniche attended.
- Thor Valentin Aakjr Olesen Nielsen attended.
- William Diedrichsen Marstrand attended.

Meeting pins:

- We have agreed to make shared decisions and planning in the first hour both Tuesday and Thursday.
- Thor will look at Jacobs pull request and check Phase and Criteria. Thor will check Phase and Criteria, implement Tag and Entry entities and fix Foreign Key functionality in model entities using ICollections for lists and dictionaries with objects that cannot be saved directly in the database. Remember to write System Design strategy before repositories. Finally he will make tests for storage repositories and implement repositories based on these.
- Talk about branching methodology. Issue can be seen in the branch "EntityRefraction" holding too big merge conflicts.
- Dennis will write on usability testing. Dennis is wrapping up on UserManagement. Need to refine documentation and tests. Dennis will present AutoMapper to the rest of the group.
- Merge AutoMapper BaseClass changes from Dennis and Jacob's pull request.
- Talk about Object Design model.
- William will first write on unit testing in testing strategy. He will finish PaperManagement. Then work on ExportManagement implementation and testing. Will begin mockups for GUI.
- Jacob will write his integration testing strategy. Jacob would like to work on functionality used to search through criteria. Jacob wants to work on search functionalities using criteria.

Done:

- Implemented AutoMapper with profiles for each subsystem.
- Wrote testing strategies.

- Fixed stored entities and foreign key functionality.
- Added Tag and Entry entity used for BibTex parsing
- Refactored and cleaned up code.
- Bug fixing.
- UserManagement tested and documented.

Sprint Planning:

- UserManagement to be done.
- PaperManagement to be done (not ExportManagement).
- Storage repositories to be done.
- Adapter between WebApi and Main Handler to be done.
- MainHandler and AutoSys Server need to be finished next week.
- Start looking at UI.

Design choices:

- **Export**: William would like to know how we import and export bibtex files and protocols. William suggests that we store bibtex files on computer as text file, we parse them in program as Paper objects and Paper will hold references to resources on the computer. This allows to find resource on computer with key from Paper. Pdf file is sent as HttpRequest as DataStream. The tricky part is to update the pdf in the directory when the bibtex is updated. However, this is assumed non vital.
- **Import**: Should BibTex parser be customized when parsing or should it have a default setup? Do we assume that the bibtex file is already imported from the start? Blue Client sends raw bibtex file to Server, this should be parsed into database. We need to take raw bibtex file and save as Papers. Blue team will support UI to import bibtex file to database. They need to do this with user defined tags outside the server. We accept bibtex syntax but could allow input like ”Retard” instead of ”Article”. Suggestion is to compare bibtex file to default tags in system. If no match it will not become invalid but just save the bibtex file with this info. Undefined tags will be defined as new bibtex tags in the bibtex parser. All well defined is saved as papers in database. All undefined tags will be saves as new bibtex entries in the database. Need to save new Tags as strings in the database. This allows to search for things in the database with bibtex tags that only exist. Field entry type (Article) can just be chosen, true false. Fields that do not exist on all papers, system needs to know if field exist. It can either compare with existing list of fields in

database. Or it can look up in database with Papers (could potentially be a big list). The field type could also have a reference to papers with this type. Solution is to have a Tag entity holding all fields as strings retrieved from id references in Paper.

- Decision: entries and fields are saved in the database and mapped with the Paper. Paper will have a entry type (Article), list of fields (enums restrict new fields) and resource reference int (used to find pdf). We will no longer make checks in the bibtex file but only import the whole file.
- Our system becomes more flexible by allowing users to upload bibtex files with custom tags that fit their Research but is not generally found in bibtex format. Tags should be stored in the database.
- New entities: Entry with string EntryType (required) e.g. History article, List of Paper ids and Tag with string TagType, list of Paper ids
-