Final Design Document

Version 4

Fontys University of Applied Science Software

Project Plan Individual project:

Tutor: Tim Kurvers & Maja pesic

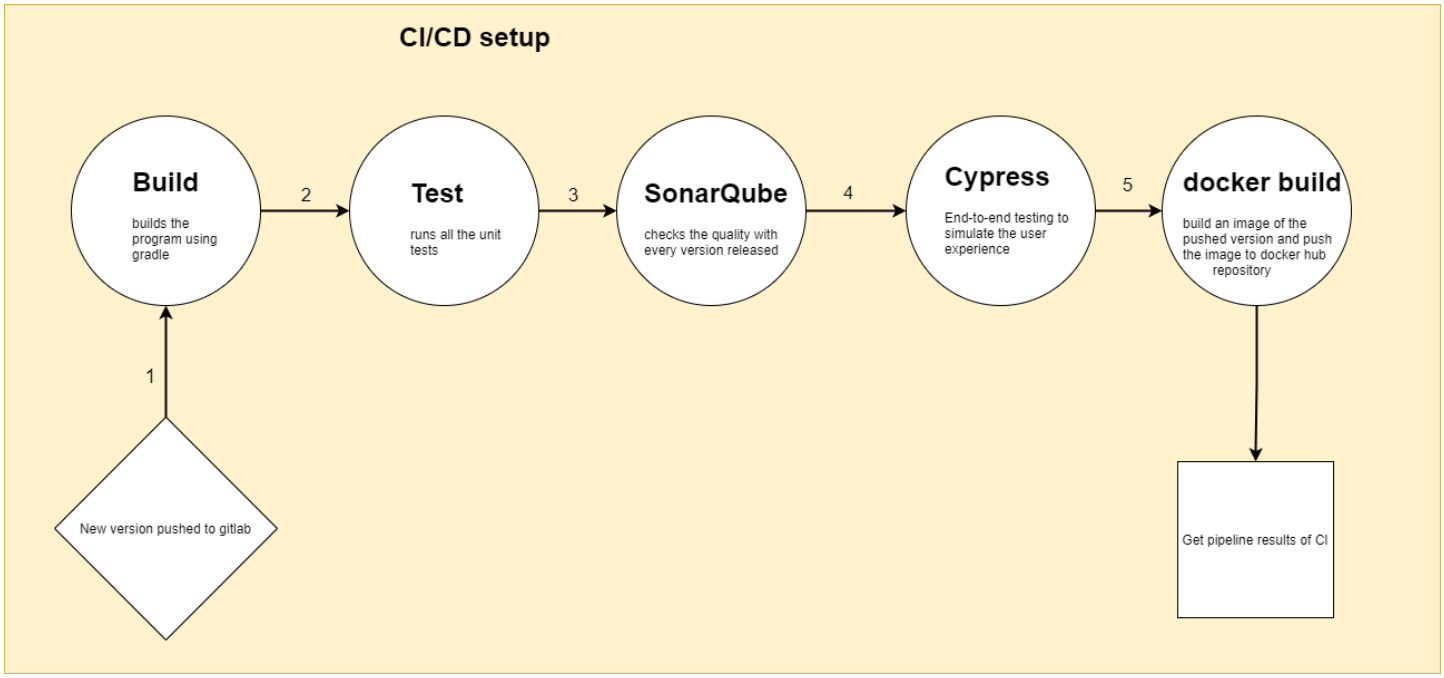
Student: Mohammed Al Harbi

4089553

Contents

CI setup

The following is the diagram that shows how CI is set up for Henaknowledge:



Design decisions

I read a blog to help me get more insights about what possible design decisions I might take and what they are in first place. I understood that they are 3 main points I need to consider while designing, my previous experience, my references and my intuition. All three combined would make that sound in my head that tells me to take a specific design based on them. Thus, I decided to ask actively for feedback on the UX for frontends I develop to have good experience help me take better decisions in the feature for any kind of project I might be working on.

Ref. (Shir, 2018)

I showed one of the teachers (Roopali Gupta) the Henaknowledge platform (Sprint 4/ Version 1.0.2 more information shown in the versioning table) and received very useful feedback regarding the design decision that were taken poorly and how could I improve them.

- Here is a list of the feedback I received from the teacher:

1- Landing page can be profile for now
2- align either everything in login in the center or all left
3- text under Henaknowledge platform should be in one line to make it consistent with the title
4- by default the list of teachers or students should follow an order for example an alphabetical order so that users
can eventually find who they look for
5- fire, banned are strong organizational words, use more positive ones. remove, access has been removed.
6- code should be generated only after clicking on a redeem points point for instance.
7- check whether a teacher can have multiple specializations and accordingly to it choose if when creating a teacher to use multiple
checkboxes or a dropdown to specify the specialization(s)
8- have a sort option to allow the student to sort teachers in particular order
9- font of search should be a little bigger so that the student can see what they searched for properly
10- improve the structure of the auto generated email, for example to have this is an autogenerated email please do not reply ..etc
11- change the words used in banning the user from Ban to something like your access have been removed if you think
it was a mistake please contact blabla@Gmail.com ..etc

- An example of a decision I took in terms of designing:

  - After user enters wrong username or password I notify the user with a text and Notification popup on the left of the screen.

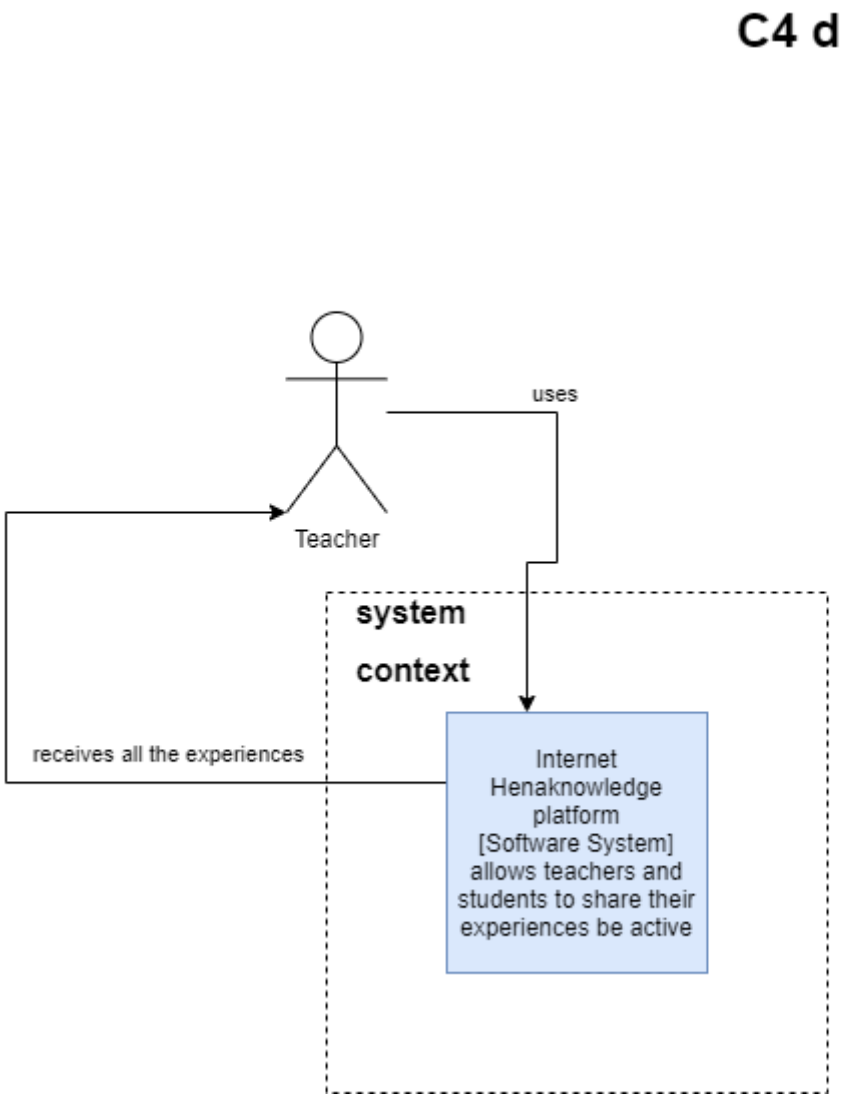
  Link: https://www.npmjs.com/package/react-notifications

DOT framework methods used:

- Document analysis
- Peer review
- Prototyping

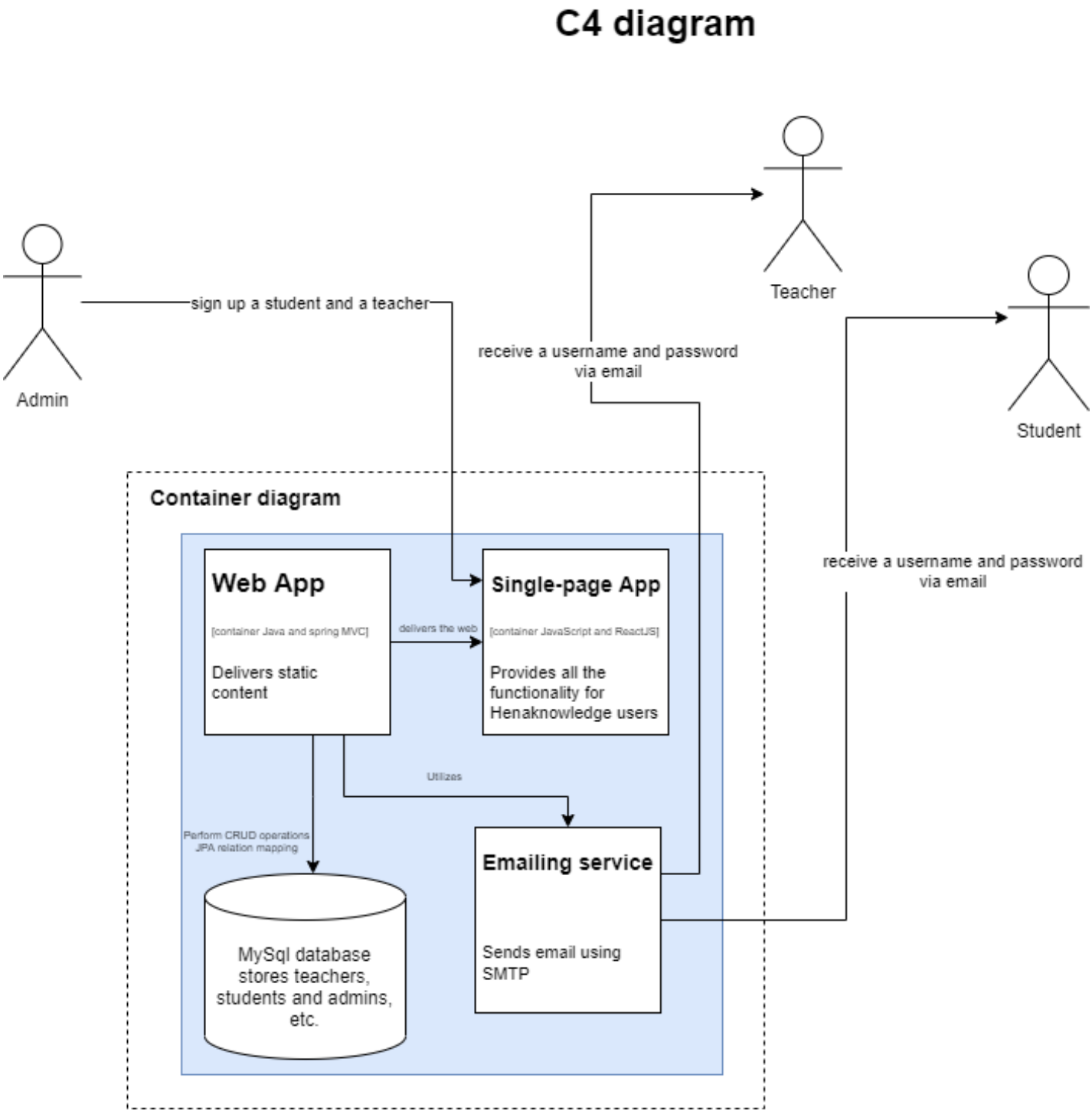
C4 Diagram

System context: C1



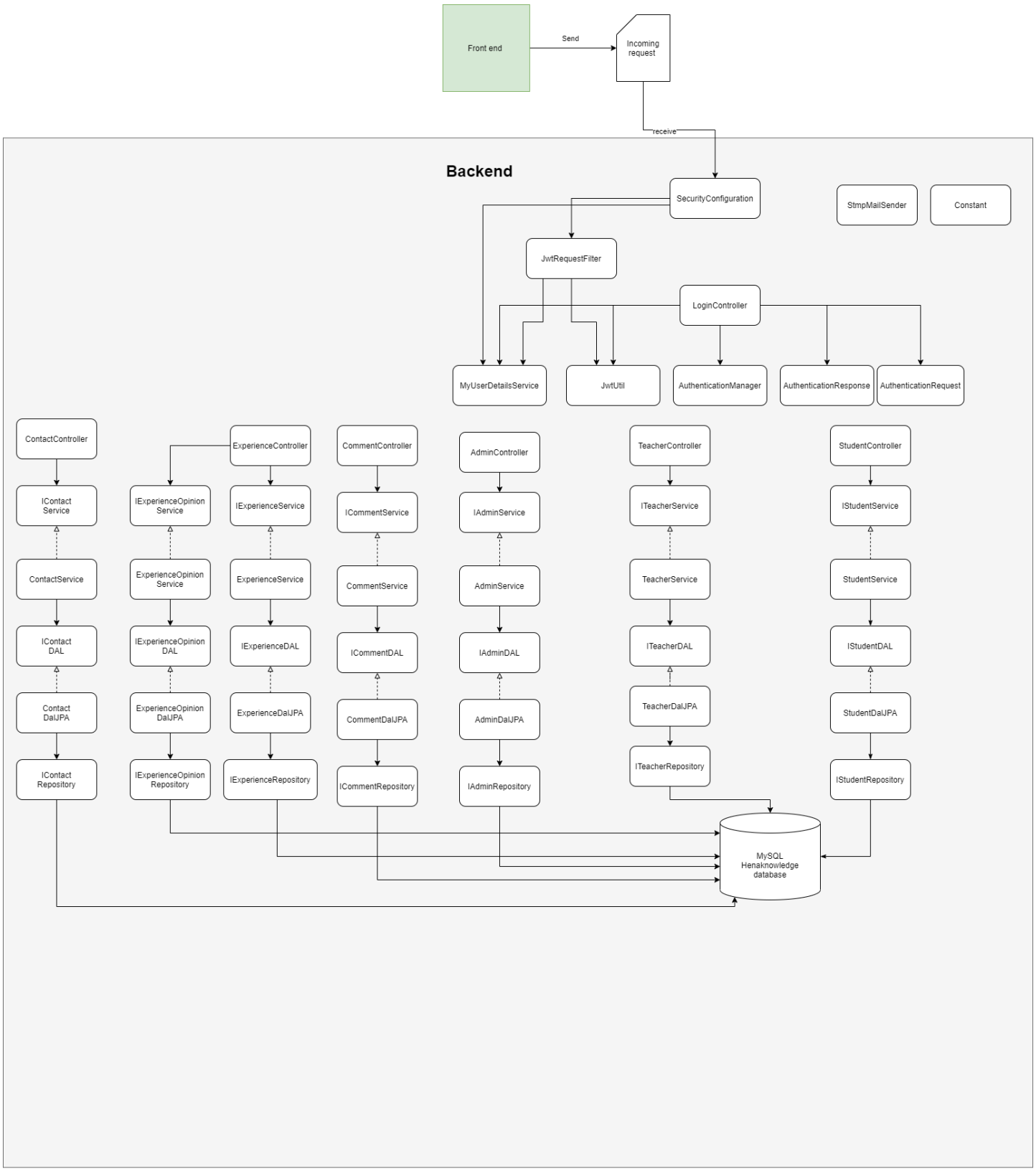
This is the first level in the C4 diagram, here the platform Henaknowledge is being used by the user profile Teacher. For instance if the teacher requested to check out all shared experiences, the platform is going to send all experiences via REST api.

Container : C2



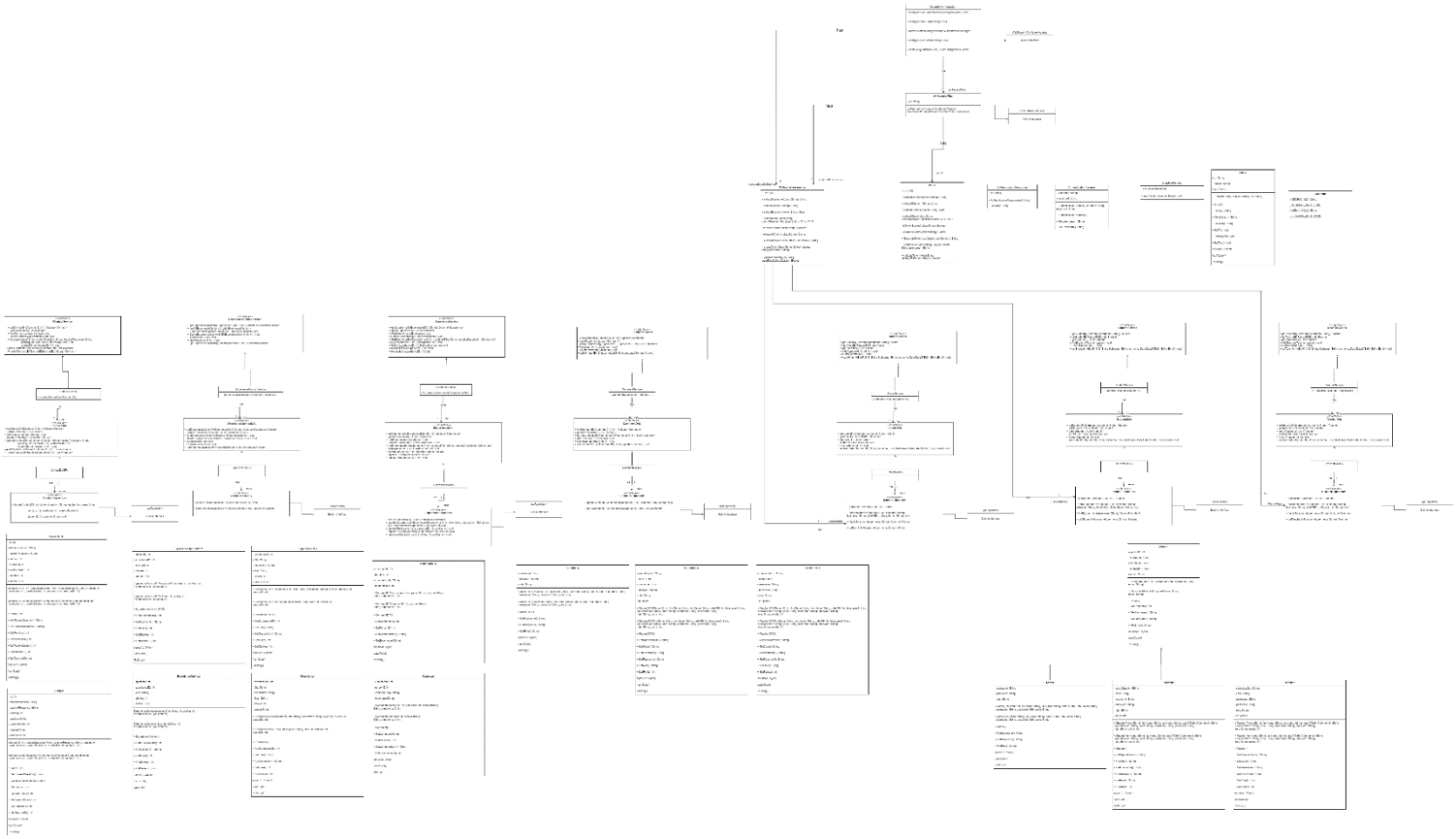
This is the second level in the C4 diagram, here the platform Henaknowledge is being used by 3 user profiles; Admin, Student and Teacher. In this level the depth has been increased and now we see more clear picture into what is happening in the platform itself. For instance, to be able to register a new student, the admin needs to communicate with a single-page app (react frontend) to provide enough student details. While the front gets its own static content from the Web App (Spring boot). In addition, once signing up process has been finished the WebApp communicate with an emailing service (google email) using SMTP for sending credentials and automated emails to the users. Finally, the information of the users are being saved into a database when is linked to the Web App that delivers the static content.

Component : C3



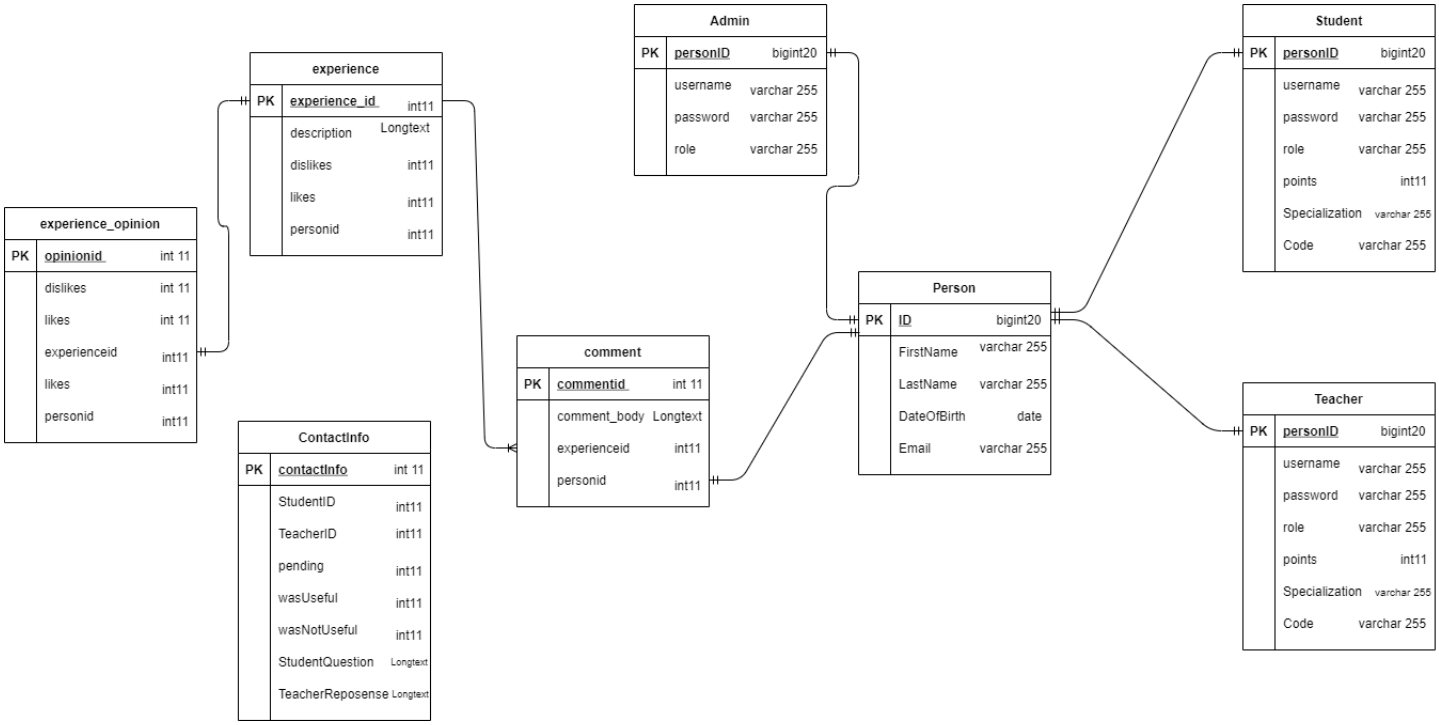
This is the third level in the C4 diagram, Here we zoomed into Web app part and the diagram above explains what classes the spring boot uses starting from security and filtering requests to controllers that uses services that is connected to repositories which are linked to the main database.

Class diagram : C4



This is the fourth level in the C4 diagram, Here we zoomed into the actual classes and interfaces that are used in coding. Pretty much a UML diagram.

ER diagram



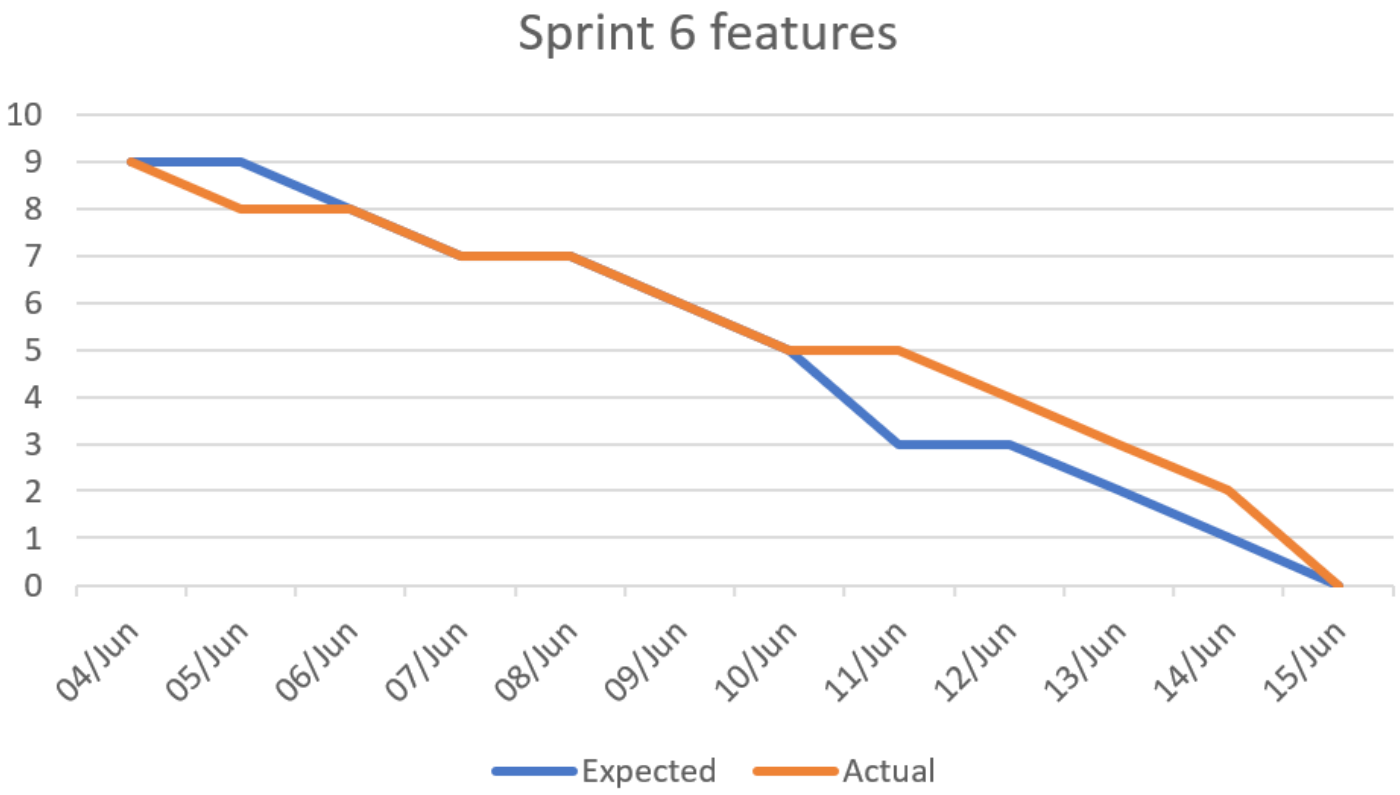
Versioning table

Sprint/Version	Functionality/ differences
Sprint 2 / Version 1.0.0	1- Crud operations(JPA) 2- no login 3- no authorization 4- no authentication.
Sprint 3/ Version 1.0.1	1- Crud operations(JPA), 2- login, authentication, 3- no authorization, 4- no jwt.
Sprint 4/ Version 1.0.2	1- Crud operations(JPA),

	2- login, 3- authentication, 4- authorization, 5- jwt generate, 6- extract information from jwt in frontend , 7- email sending.
Sprint 5/ Version 1.0.3	1- Crud operations(JPA), 2- login, 3- authentication, 4- authorization, 5- jwt generate, 6- extract information from jwt in frontend , 7- email sending, 8- sharing experiences, 9- reading other users experiences, 10- like, dislike them and comment on them.
Sprint 6/ Version 1.0.4	1- Crud operations(JPA), 2- login, 3- authentication, 4- authorization, 5- jwt generate, 6- extract information from jwt in frontend , 7- email sending, 8- sharing experiences, 9- reading other users experiences, 10- like, dislike them and comment on them 11- Points can be redeemed and earned, 12- Contact feature where students can ask teachers and teachers answer,

	13- Dash board for setting up points earned, 14- Dash board for redeeming points, 15- Data visualization to display the activity of Henaknowledge users, 16- Websockets Henaknowledge global chat

Sprint 6 burn-down chart



The expected number of functionalities to implement this sprint was 9, at the end of the sprint I managed to finish all features. Experiences, likes, dislikes and comments bug fixes, in addition to redeeming feature, contact feature, dash boards for admins and data visualization of users where asking questions answers or sharing experiences. A global

henaknowledge chat feature is also implemented. Overall, since this is the last sprint, all features and user-stories are done.

ALL REFERENCES

Npm: React-notifications. (n.d.). Npm.

https://www.npmjs.com/package/react-notifications

Shir, G. (2020, October 16). *How Do You Make Design Decisions?* Marvel Blog.

https://marvelapp.com/blog/make-design-decisions/