Team 22:

- Jaan Jaerving - KT: 310191-3539
- Snorri Steinn Stefánsson Thors - KT: 300895-2639 **Presenting**
- Steinunn Ósk Axelsdóttir - KT: 210688-2699
- Valbjörn Jón Valbjörnsson - KT: 100594-2779

Class Diagram and Project Structure

Documentation

Currently pretty much all of the project is missing meaningful JavaDoc and comments. This makes it difficult to judge functionality at a glance as we get no on-hover information or in function explanations.

Controller classes

In general we are going to be assessing the functionality that is actually present (e.g. it is not particularly meaningful to criticize the current implementation of the `AdController` as implementation is minimal.) It is difficult to make a judgment on whether missing functionality is missing because it is still TODO or it is no longer in scope.

The implementation of HomeController.java

Upon initial inspection due to the somewhat shared nature of how accounts work some functionality for `User` and `Company` entities has been moved into a new class. Specifically the `viewUser` and `viewCompany` are defined as being part of their respective controllers in the class diagram but are actually handled by a new Controller class called `HomeController`.

- It may be a better idea to split up login functionality of users and companies.

The lack of implementation of the ApplicationController.java

This controller has seemingly been removed and the only functionality that appears to remain is the ability to reroute to `makeApplication`, this is now instead defined in the `UserController`.

- Are applications important enough to have their own controller or will all of this be handled user side? Can any other entity even make applications?

Service classes

Once again here we are going to only comment on the Service classes that are present. This means that no thought is going to be given to the implementation of the planned design of the `ApplicationService` or the `AdService`.

UserService and CompanyService

These are generally implemented as described but there is a point to be made about how logins are handled that we will touch on in the security section later in this review.

Repositories

Same deal, only reviewing what exists.

UserRepository and CompanyRepository

These are also mostly as described in the class diagram.

- An exception to this would be that the diagram describes login as handled by the repositories. This would however be a strange implementation (how would a database handle logins?) So it makes sense that it was moved.
- It is possibly good to mention here that some definitions made by the repository classes are unnecessary. An example of this would be the `save` methods, JPA repositories provide these by default so explicit definitions could effectively be removed.

Entities

- There are some basic discrepancies in the naming scheme of the entity properties. This is not particularly important though, mostly it is possible to understand what is meant.
- The `ID` property of some entities seems to be missing or not properly generated. This is strange as sequential generation is used for the `Account` entity but this strategy is then not used elsewhere…

Ad

- The implementation of the `Ad` entity seems somewhat incomplete in comparison to the diagram.
- It is mentioned that there is a problem creating `ID` values for this entity - Why is a standard sequential ID not sufficient for this entity?

Application

Similar question for the `Application` entity as with the `Ad` entity, this time it has an `ID` property but it is not automatically generated. Is there a particular reason or strategy for how these `ID` properties are implemented?

The implementation of the PDF property for the Ad and User entities.

Currently this is stored as a simple String, presumably this would then be a `URL` to some stored pdf. It could be more interesting to attempt to implement actual storage of these files (it is a bit optimistic to expect users to actively use web hosting for their documents.) One possible way to do this would be to use `input=file` in the template and then store the PDF as `Base64` (at least if real file storage is not appealing.)

Other diagrams and documentation.

Because implementation is still in progress it is difficult to assess whether the diagrams match how the project functions.

- Currently it is not possible to assess the sequence diagram at all, this fails immediately.
- The implementation of the login system matches what is described in the Activity and State Machine diagrams but we cannot assess further.

The ideas presented in the Vision Document are appealing. There is a severe lack of defined salary ranges for jobs advertised in Iceland.

Security concerns

These generally don't matter for a student project but can be good to have in mind in terms of best practices and problems that are introduced when a project is made accessible to the web.

- As mentioned earlier in the service classes section there is a slight concern with how logins are handled. It seems that login comparisons are always made with plain-text passwords. This also indicates that they are in fact stored as plain-text. It would be wise to implement some minimal hashing such that if there is any kind of database leak the real passwords of the users are not available to malicious agents.
- Another possible concern is that there is currently no session checking for the `makeAd` and `makeApplication` routes. It is somewhat clear that these are not yet done, but it is nonetheless good to remember that there should be some kind of protection for these routes rather than allowing users to route to them directly. (The lack of a visual button does not defend a route on the internet.)

If PDFs are ever accepted as files:

- If the platform ever allows direct upload of files they will need to be validated for:
  - Size
  - Mimetype
  - ...