

Personal Portfolio

Ingrid Skar, n9808205

Table of Content

Release 1	3
Trello board (scrum board)	3
Burndown charts	4
Minutes	4
ER-diagram	5
Customer Database	6
Release 2	7
Automated Unit Tests	7
Design and Architecture	8
Acceptance Tests	9
Release and sprint plan	9
Source Control	9

Release 1

During the project development to the completion of the application - The 28 Delivery - my main responsibility has been making sure the process follows the scrum principles, as well as contributing to the coding part of the final application.

The artifacts of my contribution in release 1 strongly reflects this. As I had some experience using scrum principles during development processes from prior projects, I subsequently used this during this project. As my experience are based on smaller projects and Scrum is a framework which should be adapted to each particular project, I did quite a lot of research on how we could make use of the principles in our process to achieve best possible outcome. Initially I also had faced some difficulties contributing to the coding part of the project as we decided to use Visual Studio and the .Net framework. As this is a Windows based framework - and I was working on a mac OS - I used a sustainable amount of time just researching how to get started.

Trello board (scrum board)

Early in sprint 1 I developed a Trello board, which we used as a scrum board throughout the process. This was used as a simple visualization tool of the workflow during the development, and was found very helpful when we started to fall behind schedule. I created different pools for stories to do in the project, to do in the sprint, currently working on, needs help with and completed. Using this regularly we knew exactly who was working on which task, and if a team member had faced some problems along the way. This was a good replacement of a physical scrum board as the team had problems meeting more than once a week in addition to workshops.

The trello board can be retrieved at: <https://trello.com/b/niVArIMr/iab330-the-28>

Burndown charts

During release 1 I also created burndown charts for the entire process, such that each team member could log the hours they had been working on the project in each week. The burndown charts automatically updated as the hours logged changed. This included charts for each week, each sprint and both releases. Burndown charts using story points were also created for each release.

We also agreed that this was a convenient way to track the amount of hours each member of the group used during the project. This turned out to be a quite nice asset as one of the members never contributed, which was clearly shown through the hours listed.

Hours list and burndown charts can be retrieved at:

<https://docs.google.com/spreadsheets/d/1aQbBwtI54fuDo8wtK2WFYTPkNoEEBB1htBx6rYG1fLA/edit?usp=sharing>

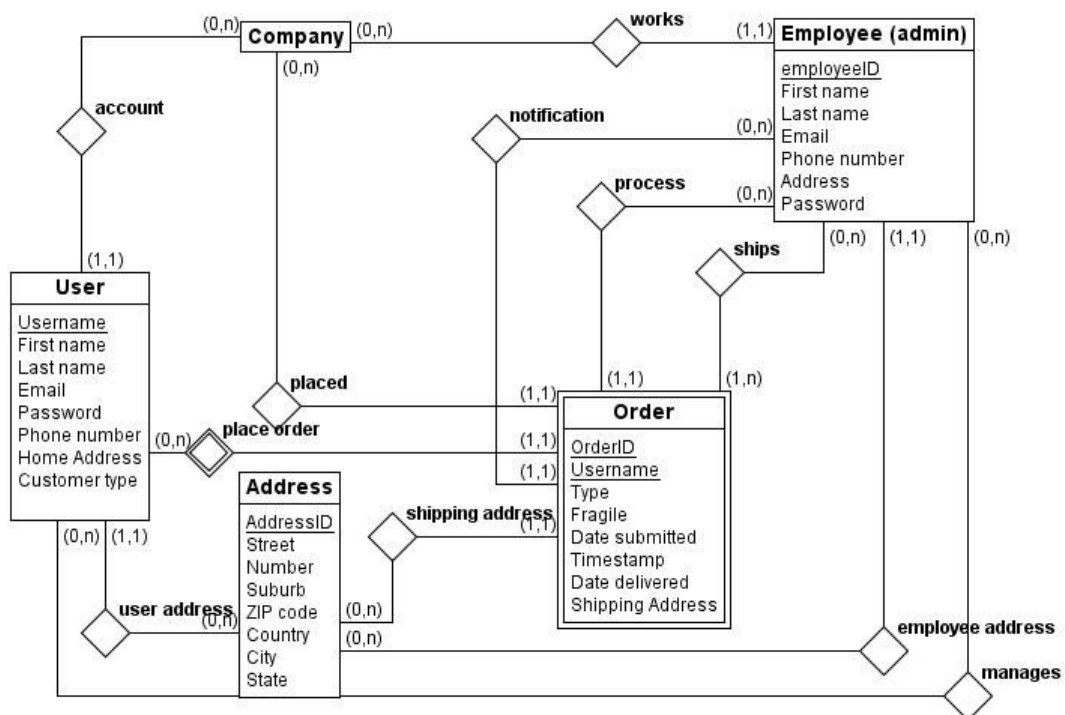
Minutes

Writing minutes for each meeting was also a contribution I felt strongly about during the process. This differentiates slightly from the principle of the scrum meetings, which initially should be informal and “casual”. As not every member could show up to every meeting this was a easy solution for all members to have the ability to know what the rest of the team has discussed. This was created primarily only for the benefit of the group, but also shows evidence of the process completed.

The meeting minutes are submitted in the ‘Documents’ folder on git, under minutes.odt.pdf

ER-diagram

We found it important to complete the database design before it was implemented into the application. I have some experience working with databases from prior units, and contributed to create the ER-diagram and the logical structure of the database. The diagram turned out to be pretty accurate, with only small changes to one of the entities. Instead of having a separate entity for 'employees' we implemented this as a subclass of 'User'. Designing a ER-diagram was very helpful while implementing the database, as it reduced redundancy as well as ensured that the data dependencies was logically stored.



Customer Database

At the end of release 1 I contributed to implement the customer database. This had been implemented at an earlier stage during the development, but not taking to consideration the normal form of the database. Due to this fact the database included a lot of redundancy which we considered very unfortunate. Especially as a main goal of our application is the ability to modify the database; by insertion, deletion and updation. Without normalization of the database this becomes difficult to handle without data loss. I also contributed to implement the site for register a new customer.

The 28 Delivery

[Home](#)[About](#)[Contact](#)[Order](#)

[Register](#)[Log in](#)

Register

Create a new account

First name

Last name

Email

Password

Confirm password

Phone number

Street address

Suburb

Postal code

State

Country

Release 2

Automated Unit Tests

During release 2, my main responsibility was the testing of the application. I had little to no experience writing automated unit test based on web applications and databases. After researching how this could be implemented to provide the team most value, I decided to write unit tests both using the Entity Unit Testing Framework and Selenium Unit Test Framework. Subsequently, writing automated test for both the database and the web application.

The unit tests written using the Entity Unit Testing Framework are basic tests on our database tables to ensure us it's working as we expect it to do. We insert data into the tables, and subsequently check if something has been added etc.

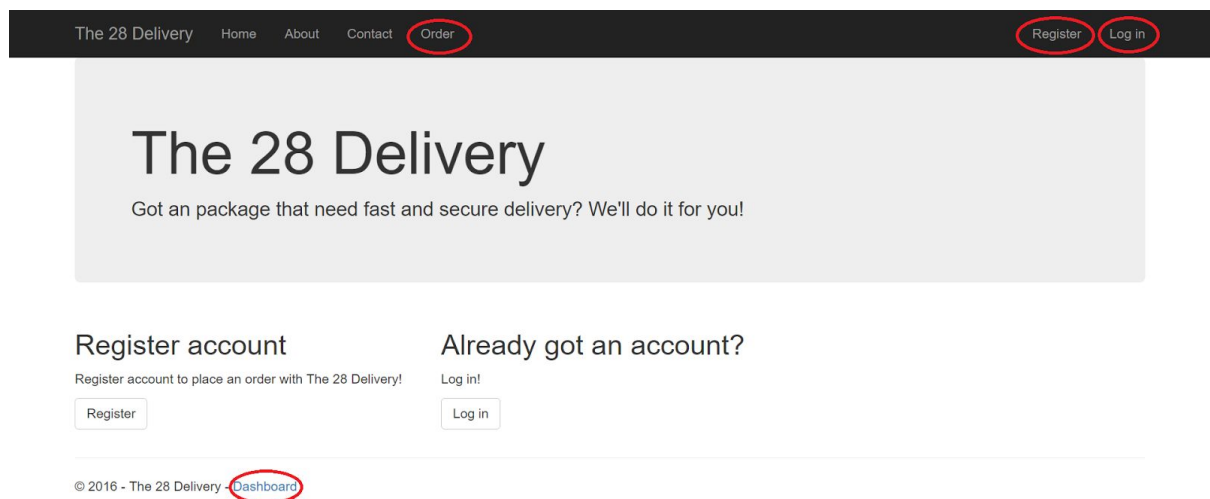
We also decided to write pretty basic test using the Selenium Framework, ensuring us that the site acted as we expected it to do while navigating around and testing the functionalities.

The unit tests are submitted as `processorTest.cs` and `seleniumTest.cs` under `Team28DeliveryTests2` on git

Design and Architecture

During the second release I also worked on the design of the application and added some small features such as the Contact page. We wanted the design to be simple and minimalistic, as it should be easy for the users to use. While designing the application we found it important that the users, either if it's a customer or administrator, finds it easy to navigate between the different views and understand the functionality of each element on the site. A element that looks like a button should act like a button, and have the same design on all pages.

The architecture of the application is also carefully planned. A customer and an administrator of the application has different use of advice, and will therefore have different access permits. To solve this we created a dashboard, only accessible for the administrator, deliberately located away from the customers' area of functionality. All user based functionality is logically placed in the navigation bar with explanatory text, in effort to make the application as intuitive and user friendly as possible.



Acceptance Tests

I also completed acceptance tests for the application based on test cases and acceptance criteria written in the first release. These tests focused on whether the solution provided works for the user, rather than if the system crashes etc.

This was a supplement to the continuous testing completed as the features were integrated with the system. The acceptance testing of our application was very important, mainly due to our 'definition of done' - a feature is complete when the respective test case is passed.

The tests are submitted in the 'Documents' folder on git, under tests.odt.pdf

Release and sprint plan

Continuously throughout the development I have been working on the Release and sprint plan. As our process follows the principles of scrum these plans continuously change as the development processes. Before each release and each sprint we walk through what we would like to achieve, and which stories to include for this to happen. As scrum - and therefore our process - is agile, changes to the initial plan has been made as result on how we have been processing. These changes has been documented in the release and sprint plan document.

If there have been any changes to the original plan, each upcoming sprint has an explanatory text describing how the plan has changed and why.

The release and sprint plans are submitted in the 'Documents' folder on git, under ReleaseandSprintPlans.pdf

Source Control

A final artifact of my contribution towards the completion of the project is source control and git. I had some prior experience using git from prior projects, which was valuable due to the fact that some of the members of the group had any. Even though I had used git before, this was primarily based on smaller projects and it turned out to be a somewhat new experience for me. As we were working on the same file simultaneously, we faced the problem of merge conflicts quite often.