MMSE - Project Report

Nicolas Goutay, Stefano Alberigo

User Stories

Time estimation is on a scale of **1** (short) to **5** (long).
The implemented user stories are marked with a ✓

1. ✓ *(Sample)* **Login** (3 + 3[1]): At startup & when the system starts, the user interface will present a login screen. The user can now log in using its user name and password. After verification the user is presented with the set of actions it can perform.
2. ✓**Right Insurance Company** (1): At reception of a claim, the system will check whether the claimant is insured by the organisation or not.
3. ✓**Claim Rejection** (1): At reception of a claim, the system will inform the claimant that his claim is rejected if he is not insured by the organization.
4. ✓**Claim evaluation** (3): At reception of a claim, the system should evaluate the severity of a claim, according to the price of the car, the cost of damage, the previous history of accident and the personal opinion of an employee.
5. ✓**Form sending** (2): After evaluation, the system will send the claimant the relevant form according to the severity of the claim via Email, Webform or SMS.
6. ✓**Form completeness** (1): After the forms are returned, the system will check for their completeness.
7. ✓**Claim registration** (4): If the forms are complete, the system will register the claim in the Claim Management system.
8. ✓**Form update** (2): If the forms are not complete, the system will ask the claimant to update them and send them again via Email, Webform or SMS.
9. ✓**Insurance check** (2): As an employee of CD, I want to check if the insurance of the claimant covers the claim. If not, the claim is refused.

10. ✓**Garage contact** (1): As an employee of CD, I want to contact the garage for a claim, via Email, Webform or SMS
11. ✓**Damage history check** (4): As an employee of CD, I want to check damage history for a complex claim.
12. ✓**Claim decision** (2): As a claim handler A, I want to make a decision about a claim : OK or Not-OK.
13. ✓**Decision letter** (3): As an employee of CD, I want to write and send a letter to the claimant, in order to tell him the company's decision about his claim.
14. ✓**Decision letter template** (2): The system should propose a template for the decision letter, including the client's name, surname, and the decision for the claim. Others details can be added by the employee.
15. ✓**Logout** (3): When the system stops or a logged-in user wants to log out, the user interface will present the Login screen and no user can perform any operations unless logged in.
16. ✓**Claim search** (4): As an employee, I want to search the existing claims by the name of the claimant.
17. ✓**Claim status** (4): As an employee, I want to see the current status of a claim.
18. ✓**Customer search** (3): As an employee, I want to search the existing clients by their names.
19. ✓**Claim record keeping** (3): As an employee, I want to see all past claims, their category, and other informations.
20. ✓**Client adding** (2): At the reception of a claim, if the client is not insured by the company, the employee has the option to register it.
21. ✓**Claim processing notification** (4): As an employee of CD, I want to be notified when I can evaluate the severity of a claim. When I dismiss it, the notification disappears both on my session and on any other session.
22. ✓**Claim decision notification** (2): As an employee of CD with a rank equal to A, I want to be notified when I can make a decision about a claim. When I dismiss it, the notification disappears both on my session and on any other session.
23. ✓**Letter sending notification** (2): As an employee of CD, I want to be notified when I can send a decision letter. When I dismiss it, the notification disappears both on my session and on any other session.
24. ✓**Client record keeping** (3): As an employee, I want to see all clients, their

claims and other informations.

25. ✓**Insurance choice** (2): As an employee, I want to be able to pick an insurance between all existing ones at the registration of a new client
26. **Payment order issuing** (4): As an employee of the Finance department, I want to issue a payment order for a claimant.
27. **Payments record keeping** (3): As an employee of the Finance department, I want to see all past payments, and their recipient, amount, and date.
28. **Payment Notification** (2): As an employee of the Finance Department, I want to be notified when a payment needs to be made.
29. **User management** (5): As an administrator, I want to be able to add, remove and modify existing users.
30. **User record keeping** (5): As an administrator, I want to be able to see the actions done by a certain user.
31. **Undo actions** (5): As an administrator, I want to be able to undo an action done by a user (such as the decision on a claim), on his behalf.
32. **Reset password** (2): As a user, I want to be able to reset my passord if I have forgotten it. This request should be validated by an administrator.
33. **Contact a client** (3): As a user, I want to be able to contact a client directly from the Seach User screen. The contact can be via WebForm, email or SMS.
34. **Insurance upgrade** (2): As an employee, I want to be able to propose the client a new insurance if I judge that his/her is not fitted to his needs.
35. **Client modification** (2): As an employee, I want to be able to change the informations about a client (new adress, new email).

First Release

The first release will contain the core functionnalities of the system, as well as the GUI. Users will be able to add clients, claims, process and decide claims, write decision letters, search through claims and clients. The login system will be implemented.
However, the payment system will not be implemented, nor will be the administrative one.
This release will allow employees to familiarize with the system, and to provide feedback on these core functionnalities. The GUI may be modified later on if the

feedback is sufficient.

Iteration 1

Iteration 1 should include the following user stories :

- Login
- Claim evaluation
- Form sending
- Form completeness
- Form update
- Claim registration
- Claim decision
- Logout
- Client adding
- Insurance choice

Iteration 2

Iteration 2 should include the following user stories :

- Claim search
- Claim status
- Customer search
- Customer record keeping
- Claim rejection

Iteration 3

Iteration 3 should include the following user stories :

- Right insurance company
- Claim rejection
- Insurance check
- Garage contact
- Damage history check
- Decision letter

- Decision letter template
- Claim processing notification
- Claim decision notification
- Letter sending notification

Metaphor

The system works like a logical series of counters, like in an administration, where each counter helps the user in accomplishing a particular task. The first counter is the *registration counter*, where the user hands in the claim's details and the claim is *registered*. The second counter is the *processing counter*, where the user just is told all the required tasks to do and helped in doing them (the counterman shows him all the relevant details for his choice). At the end, the claim is *processed*. The last counter is the *decision counter*, where the user is shown all relevant details and have to make a decision. Every counterman can raise its hand (it's a *notification*) when he is needed, so as to catch the user's attention.

Pair programming

It was, for both of us, our first experience of pair programming.
We agreed on some rules (only one computer, 1-hour shifts, working complete days when possible) and it went quite well. We both had some Java experience from previous projects, so the flow was quite fluid.

The estimations were, overall, correct. However, the construction of the GUI, at the very beginning, has been much more time-consuming than expected : both of us had to learn again how Swing and AWT work, and how to manage events between multiple panels. We also had some trouble in finding (and implementing) the Client and Claim class, for they had to have some properties in order to allow us to design the system without files or a database.

As these tasks were at the very beginning of the project, we felt pessimistic because we thought we were late. This showed us it is important to actually think things through when doing time estimations.

Stand-up meetings

Day 1

Last day : Nothing

Today :
- Crash course in Java GUI development
- Design and implementation of the skeleton of the GUI
- Login user story

Day 2

Last day : We managed to build the skeletton of the UI, but it took us moch longer than expected. We noticed that we could use an Observer pattern when managing user interactions. The login user story is not finished yet.

Today :
- Refactor the GUI code
- Finish login
- All user stories from iteration 1

Day 3

Last day : Iteration 1 is implemented, but some refactoring has to be done. We had trouble in imagining a system without file IO of databases, but we achieved it. The code began to be sloppy on the end, so we have to double-check it.

Today :
- Refactoring
– Use an Observer pattern where applicable
– Check if standards were respected.
– Create packages to keep the code clear
- Claim search / status user story

Day 4

Last day : We had to modify some things in the Claim class, and some tests failed, so we had to revise some pieces of code. No major difficulty, the project is on tracks.

Today :
- All stories from iteration 2
- Refactoring iteration 2 code

Day 5

Last day : No major difficulties. Refactoring went well, and time estimations were quite accurate.

Today :
- All stories from iteration 3 (should be easy, mostly GUI manipulation)
- Refactoring
– Looking for inheritance factorization
– Looking for Observer patterns

Day 6

Last Day : Iteration 3 is finished. We did find some code factorization that reduced significantly our GUI. The system is up and running.

Today :
- Refactoring
- Acceptance tests

---

1. The additional time estimation is about the reflexion on the GUI, and it s basic implementation ↩