Project CS - Course Report
Klarna E-commerce Framework

Christian Rennerskog, Shahzad Gul, Nicklas Nordenmark
Manzoor Ahmad Mubashir, Mikael Nordström, Kim Honkaniemi
Tanvir Ahmad, Yujuan Zou, Daniel Widgren

January 15, 2010
Abstract

There are large numbers of e-commerce frameworks in the market but most of them lack the ability to distribute and scale. This project aims to develop a ready-to-use, highly customizable and stable e-commerce framework, written in Erlang, that provides the functionalities other frameworks do not. However, this product is not about competing with other frameworks but rather, experimenting with new cool technologies at hand and demonstrating it to the prospective developers who are new to these tools.
Contents

1 Introduction 3

2 Resources 4
  2.1 Hardware ........................................ 4
  2.2 Software ........................................ 4
  2.3 Manpower ....................................... 4
  2.4 Literature ...................................... 5
  2.5 Environment .................................... 6

3 Project Methodology and Organization 7
  3.1 Organization ..................................... 7
  3.2 Scrum ............................................ 7
    3.2.1 Introduction to Scrum ...................... 7
    3.2.2 Roles ....................................... 7
    3.2.3 Daily Stand-up ............................. 7
    3.2.4 Sprint Planning ............................ 7
    3.2.5 Sprint Review .............................. 8
    3.2.6 Scrum Board ................................. 8
    3.2.7 Burn-down .................................. 9
  3.3 Our Implementation of Scrum .................... 9
    3.3.1 Project Backlog ............................ 9
    3.3.2 Sub-groups .................................. 9
    3.3.3 Sprints ..................................... 9
    3.3.4 Meetings .................................... 10
    3.3.5 Demos ....................................... 10
    3.3.6 Scrum Board ................................ 10
  3.4 Methods and Tools ................................ 11
    3.4.1 Whiteboards ................................. 11
    3.4.2 Redmine .................................... 11
    3.4.3 Hudson ..................................... 11
    3.4.4 Subversion .................................. 11
    3.4.5 Mailing List ................................ 11
    3.4.6 IRC .......................................... 12
  3.5 EUC conference .................................. 12
  3.6 Review .......................................... 12
  3.7 Problems ........................................ 12

4 Review 13
  4.1 Technical Review .................................. 13
    4.1.1 Opinions of the Software .................. 13
    4.1.2 Development Decisions ..................... 14
  4.2 Cooperation ...................................... 14
    4.2.1 Klarna ...................................... 14
    4.2.2 Nitrogen & Risk Development Team .......... 14
4.3 Recommendations to Next Year .................. 15
  4.3.1 Recommendations to Students ............... 15
  4.3.2 Recommendations to Course Administrators ... 15

5 Conclusion ......................................... 16

6 References and Websites ........................... 17

A Contributions ...................................... 18
  A.1 Christian Rennerskog .............................. 18
  A.2 Manzoor Ahmad Mubashir ......................... 18
  A.3 Nicklas Nordenmark .............................. 18
  A.4 Kim Honkaniemi ................................ 19
  A.5 Tanvir Ahmad .................................... 19
  A.6 Shahzad Gul ..................................... 20
  A.7 Yujuan Zou ....................................... 20
  A.8 Daniel Widgren .................................. 21
  A.9 Michael Nordström ............................... 21
1 Introduction

The goal of this project was to build a ready-to-use, highly customizable and stable e-commerce framework that combined the functionality of a normal web framework with the features a distributed key-value store provides, together with the scalability and stability that Erlang offers.

There are several e-commerce frameworks today that are also open-source. For us, the goal was never to compete with them. Rather, we wanted to try new technologies and see if we could develop a working e-commerce system using them.

For this project, we used the Nitrogen Web Framework [1]. This is a new event-driven web framework for Erlang [2] which is quick to learn and with lots of web 2.0 capabilities. As most traditional databases are relational and non-distributed, the newly developed Riak [3], our underlying data store, was a decentralized key-value store that could scale well, and in many ways, be a more natural fit for rapid web development than traditional RDBMS.

In the software development of this project, we used Scrum [4]. Scrum offers the opportunity to quickly produce working prototypes and constantly gives a good overview about the project’s progress. As Scrum is an iterative process, when changes are made, a lot of the code does not need to be deprecated but instead can be reused.

We were nine Masters students in the Computer Science department of Uppsala University. The project was a cooperation between Uppsala University and Klarna [5], and was part of the course (Project CS) during the autumn term of 2009. Klarna provided us with the project outlines, concepts, and lots of help and feedback during this time. For our group, this was a new experience working in a large group doing a real-time project work which will, hopefully, turn out to be a good and valuable experience for our future educational life.
2 Resources

2.1 Hardware
In total, nine desktop computers and one server was used to develop the software in this project. The computers had the required specifications for running all the necessary applications needed for the project.

2.2 Software
The following software was used during the project:

Operating systems:
- Ubuntu Linux 9.04 & 9.10 - Running on the desktop computers and server

Development tools:
- gedit - Preferred text editor by some
- Emacs with Erlang mode - Extensive text editor used by others

Server services:
- Redmine - Project management and bug-tracking tool [6]
- Subversion - Version control system [7]
- mochiweb - Lightweight HTTP server for running Erlang code [8]
- Erlang - Our main programming language for this project
- Nitrogen - Web framework for Erlang
- Apache2 - Web server running Redmine among other things
- MySQL - Relational database used by Redmine
- Riak - Decentralized key-value store, used by our application
- Hudson - Continuous integration tool [9]

2.3 Manpower
The project group consisted in a total of nine members (initially ten) with four international students and five Swedish students. Some of us had considerable experience in web development including a few having experience in making e-commerce systems even. But no one had worked in large projects before. From the beginning, we let each other choose the areas he or she wished to work in. We always kept a door open for us to switch the areas we are working on, but
most of us chose to continuing working in the same area throughout the project with only minor variations.

Early in the project, we had discussions so that we could get familiar with each other and then, certain rules were set so that our work can be managed in an organized way. When speaking at the meetings or about work related issues, we tried to speak English as much as possible. However, this did not work out very well. Often, while working in small groups, we tend to speak the language that was used by those in that group. This left out people from discussions that they might have found interesting and would have liked to participate in. During the project, we had several social events outside work to give ourselves a chance to know each other better. We also played card games together during regular coffee breaks. This led to a good working spirit and people were not afraid of speaking to each other.

2.4 Literature

Since Erlang was the required language to use in the project, some of us bought the book *Erlang Programming*\(^1\) to use as reference guides. Instead of using the book, we used Internet resources such as the Erlang R13B Documentation, Nitrogen and Django reference at their respective home pages. Sometimes searching the web was found to be the best option for retrieving information for a particular problem.

Since the Nitrogen and RIAK database documentation, at least in the beginning, was found to be mostly insufficient, we contacted their development teams. They gave us a very good feedback and provided a lot of help in solving tricky issues that we encountered during the project.

To learn the Scrum method, we got the book *Scrum and XP from the Trenches*\(^2\). This helped us a lot in using Scrum in the right way.

\(^1\)Francesco Cesarini & Simon Thompson, O'Reilly, 2009

\(^2\)Henrik Kniberg, InfoQ, 2007
2.5 Environment

The picture above shows the room in which we spent most of our time. It was a large room with a high ceiling and large windows with a nice view to the nearby surroundings. The room space was more than enough for our needs. Next to the project room, we had a nice lunch/conference room with a refrigerator and microwaves. This was a perfect room for sprint planning, coffee breaks and so on. However, a few times we have experienced door access problem to the project room which was fixed in due time.
3 Project Methodology and Organization

3.1 Organization
As we were only nine students in our project group, we did not feel we had to split up into sub-groups. Nevertheless, we had two teams; one was responsible for the database layer and one was handling the web pages. The task of these two teams were tightly coupled in the work domain. Had we not split the team into two, we feel a lot of good communication and coordination could have been lost so this split was very necessary.

3.2 Scrum

3.2.1 Introduction to Scrum
Scrum is an agile software development method for managing complex work. The method is a iterative incremental framework where the focus is to produce result as fast as possible. The iterative cycle in Scrum is called a sprint and can be between a week and a month long. At the end of each sprint the team should have a result that can be demoed. This gives an indication of how well the projects are going.

3.2.2 Roles
In Scrum there are a few important roles. First, there is the product owner who represents the customer. The product owner decides what the team should do, create general tasks and then prioritize them. Then there is the Scrum-master whose role is to make the team follow Scrum and make sure the group meets the sprint goals. The Scrum-master is not the same as a project leader. The focus is not to decide, but rather guide in what the team should do. The last role is the team which is responsible for delivering the product in time.

3.2.3 Daily Stand-up
An important element of Scrum is the daily meetings. In agile development it is most important that all team members are aware of what everyone is doing. This way, problems are easily discovered and you could quickly find out if someone needs help in the group.

3.2.4 Sprint Planning
The sprint planning is a planning meeting where the Scrum team decides what general tasks they will be able to complete during the upcoming sprint. Large tasks are then divided by the team into smaller ones. All tasks are then prioritized and estimated how much time they will take. Everybody participates in this work.

One way to make sure everybody participates is to use planning poker. Planning poker cards are cards with different numbers on them representing time.
The number can represent working days, hours or relative numbers. Everybody in the team gets a set of cards and picks out one of them as an estimation on how much time a task will take. When everybody has picked a card, the cards are revealed and a discussion starts. The individuals with very large values have to explain and motivate their choice. If the team does not agree, they do the procedure over again until an agreement is reached. In the end, all tasks are written down on post-it notes and put up on the Scrum board.

### 3.2.5 Sprint Review

At the end of each sprint, there is a sprint review where the members get the opportunity to give feedbacks and comments. A common way is to hand out post-its where the team members write what has been good or bad. The post-its are then gathered in and presented. The bad items are then up for discussion on how the team should solve them.

The sprint review is followed by a demo where the team presents what they have accomplished so far. This gives everybody a chance to see what is going on in the project.

### 3.2.6 Scrum Board

The Scrum board gives an overview of how the sprint is going and what everybody is working on. All tasks are put up one the Scrum board in the form of post-its. The Scrum board consists of five sections:
Not checked out  Tasks that no one has worked on.

Checked out  Tasks that someone is currently working on.

Done  Tasks that are finished.

Unplanned  Tasks that have to be finished during the current sprint but was not planned at the sprint planning meeting.

Next  Tasks that should be completed in the next sprint but can be done in the current if time allows.

3.2.7 Burn-down
The burn-down is a graph that shows how much work has been done in the current sprint and how much time that remains on the tasks. The burn-down is updated every morning by the Scrum-master at the daily stand-up meeting.

3.3 Our Implementation of Scrum

3.3.1 Project Backlog
From the beginning we had a good backlog to start with. This was a list with ideas on what our product owner would like to see in this project.

3.3.2 Sub-groups
Though it is recommended that groups larger than seven people should be divided into sub-groups, we felt that this was not ideal in our project. In practice, we worked in two teams, one database team and one web team. A lot of tasks needed both parts to be active in the development and therefore the Scrum group was never divided. This also led to good communication between the teams, which was necessary for us all.

Since many of us wanted the experience as Scrum-master, we changed this role between members every sprint. Unfortunately, this led to a little chaos like no one never really got used to the role and every new sprint we had a new Scrum-master who stumbled a bit in the dark before knowing what to do.

3.3.3 Sprints
Throughout the whole project, we had two week sprints. Since we were totally inexperienced with the development method and the frameworks we used, a lot of problems were expected together with a lot of changes during a short period of time. Considering the planning and workload we had, we all feel that this was the most optimal length of a sprint.
3.3.4 Meetings

Sprint planning meetings were performed quite informally with everybody sitting in our lunch/conference room. In the beginning, these meetings could be somewhat chaotic, due to our inexperience, the lack of a good Scrum-master and the absence of a product owner at our meetings. The poker planning cards were used a lot but were a bit confusing too when everybody had different ideas of what the numbers represented.

The daily meetings that were performed standing up at a certain time at mornings, were arranged with no problem. The meetings gave good insight in what had been done the previous day and what tasks were going to be worked on the current day. It also gave a good insight in who needed help and what people were going to work together that day.

3.3.5 Demos

At the end of each sprint we had a simple, informal demo of what we had done. This was shown to the team members as well as the product owner. This worked very well because it acted as a motivation because everybody gets an overview of how far things have reached in the project.

3.3.6 Scrum Board

The Scrum board is the physical representation of the Scrum methodology. The board shows how things are structured, who does what task, and in what status a task is in. This was the place were the daily meetings were held to further give everyone a good overview of the sprint.

For us, it felt natural to use a real Scrum board instead of software which emulated the methodology. Simply because it was a lot easier to use and gave a lot better overview for everybody. The post-its tended to fall down sometimes but this was solved by having magnets for those loosely-hanging ones.

In our Checked out section we used magnet name tags to put on those tasks we were working on. This gave us nice overview on who was working on specific tasks. We also decided to add some extra sections to the Scrum board. This was to further give an overview of the sprint and the status of a task.

**Worked on** A subsection to Not checked out. If the task had been worked on but needed further work later on, it was put here.

**Sprint goal** Some general goals on what we were supposed to do this sprint.

**Deprecated** Tasks that were found not working with the rest of the system design.
3.4 Methods and Tools

3.4.1 Whiteboards
In our project room we used a total of three whiteboards. One of them worked as the Scrum board, one as a general planning board, and the last one as an post-it board for important things that needed to come to everyone’s attention. The whiteboards were of great use and put in the room so that everybody could get an good overview of them.

3.4.2 Redmine
Redmine is an open-source, web based project management and bug-tracking tool. This was set up on our server. Redmine helped us in several aspects:

- Wiki
  Redmine supports a Wiki function which we use to put up documentation and tutorials, contact list, absence list, and important announcements.

- Storing a digital version of the Scrum board for keeping history. This is done in an issue tracking tool where bugs and tasks can be posted. The tracking tool also provides a roadmap on how much we have done the current sprint.

- Track changes in the Subversion repository.

- Upload files for everybody to access.

3.4.3 Hudson
Hudson is a continuous integration tool we used to build and back up our system with. The system helped us in doing tasks that needed to be scheduled and automatically handled.

3.4.4 Subversion
We used a version control system called Subversion (SVN) for keeping track of the code and other project files. This was essential for the project and worked nicely throughout the project.

3.4.5 Mailing List
From the beginning of the project we used a Yahoo mailing list. Except mailing out important notices we could forward our answers we got from the Nitrogen & Riak development team to our group. This was a simple and good tool which was of good use.
3.4.6 IRC

In the beginning of the project, we set up a IRC channel for the team members to use. We quickly found out that this was not used at all. Since all members worked in the project room and no one ever worked from home, the tool was useless.

3.5 EUC conference

All of us agreed that we got a good experience out of the EUC conference. Besides getting some insight in Erlang development, we also got good feedback from people interested in the project.

3.6 Review

The project review had several positive impacts. During the time we prepared for the review, we ourselves could get a feeling for what we had accomplished so far and exactly what could be done with our system, and what needed improvement. The audience in the review also gave useful feedback. Unfortunately, we could have used a bit more time preparing for the review, making our system a bit more presentable.

3.7 Problems

In sprint planning, we sometimes lacked a product owner. Planning could sometimes be a bit confusing when we had problems deciding what to prioritize in the backlog, or what something actually meant. This could have been solved by always having the product owner at our sprint planning. Half way through the project, we got Richard to be there which greatly improved our meetings.

Estimating time was something that could drag our meeting times to infinity. Besides that, members could have different ideas of what the numbers meant and on many occasions, we didn’t have a clue about what problems we would be facing in doing a task. This made estimation a bit hard.

Sometimes the communication between the group members could be a bit chaotic. At one point, a member was adding stuff to the database while the other one was deleting them. Both of them were confused when they were not getting the expected results. These communication problems were to some extent due to language problems, and also due to lack of proper usage of the Redmine system as members were not checking what tasks other project members were working on.
4 Review

4.1 Technical Review

4.1.1 Opinions of the Software

Nitrogen  Nitrogen contains lots of powerful features that made development go a lot faster. On the other hand it was lacking some very important aspects.

- **Standards**
  Nitrogen does not render any real XHTML standard, some tags are not properly closed and so on.

- **Internationalization**
  Several of those components included in Nitrogen had hard-coded language in them. Not so nice when building a system with support for internationalization.

- **Templates**
  For building a non-trivial application, you have to integrate a better template system since the Nitrogen template system becomes unsustainable.

  We would still say that this is a good framework if you are using Erlang to build your web pages. Nitrogen easy creates different effects on your pages, everything from a wizards to just simple pop-up confirmations. One other of the good things in Nitrogen, is the error trapping functionality. In conclusion this was a good tool to use and very simple.

ErlyDtl  Apart from having some integration problems with Nitrogen [10], ErlyDtl [11] is nice template language. Not all tags are implemented from Django [12] but it was good enough for our needs.

Mochiweb  More then installing we didn’t use it more then just running the application. We used a old version because the newest version created some large errors for us so we had to go back many revisions before it worked again.

Riak  This was a new tool for us to work with. In the beginning, we felt that Riak was too young to do something useful with. The problem we had with Riak was that it was new and changed a lot during our project. When we started they were on 0.4 and released 0.7 one week before the end of the project. For almost every new release they changed the API and didn’t really document it in any good way. This created some extra work for us. After getting in contact with its development team and reporting some bugs and getting new versions, we were getting somewhere. Riak can be a good tool for e-commerce because of the scalability and the way it stores data. But the constant changes of the API in Riak had it downsides. One problem is that Riak doesn’t have an management system. Another problem that we had was that it were slow when
we had much items in a bucket. For example, searching and paging took a lot of
time in implementation. This is more because of bad design from our side than
of the implementation of Riak. Web development with Riak is not as trivial as
working with an RDBMS.

4.1.2 Development Decisions

CSS Framework We decided not to use any CSS framework. This was due
to the fact that we felt we didn’t have time to integrate yet another framework
and it was easy enough to write our own CSS. In future, we would probably
consider integrating some framework like Blueprint CSS.

Template framework When building a non-trivial application, Nitrogen’s
templates system became unstable. Since we wanted a system where as much
layout could be changed by our customers, we needed a suitable template sys-
tem. To get this we had to integrate Nitrogen with ErlyDtl. This worked well
until we tried to use the Nitrogen event system. We ended up with having a
mix of both Nitrogen and ErlyDtl template systems. Integrating ErlyDtl in our
system forced us to change much of our code based on the customer pages. This
took a lot of time but on the other hand it helped us a lot in creating a nice
plug-in system.

4.2 Cooperation

4.2.1 Klarna

Richard Carlsson was our main contact throughout the project. Richard has
been most helpful in every way and has really shown an interest in what we’re
doing. During the project we also got a visit from Erik Stenman (Klarna), the
man with the idea for our project.

In November, we were at EUC in Stockholm and met some of the people
working at Klarna. We made a small demo of our product to those interested
and got some feedback from them.

In December, we visited the Klarna team in their office in Stockholm. There,
we demoed our product to them and got a tour around their offices.

4.2.2 Nitrogen & Riak Development Team

After getting in contact with Rusty Klophaus, the creator of Nitrogen and cur-
cently working for the company doing Riak, we were in certain periods of the
project in contact almost every day. He helped us with many of the more dif-
ficult questions that faced us. Before the EUC we had the pleasure to meet
him.
4.3 Recommendations to Next Year

4.3.1 Recommendations to Students

• Get to know your project members.
  It is good to have a tightly coupled project group because this makes the work easier. You should plan for social events outside your work domain at the beginning of the project, have regular coffee breaks and play some sort of games (like card games) in between. These are simple and easy ways to bring the group together.

• Learn Scrum!
  Scrum is useful both in this project and later in real life. Take this opportunity to learn it. And if you feel it is a bit much more to swallow, implement it bit by bit.

• Set rules!
  If you need to set rules for your project group, do it! Rules like that you need to speak English if there are international students in your group, or that everybody should be in the project room during office hours, can help a lot.

• Plan before you do things.
  Even if Scrum lets you experiment a lot due to its iterative process, good planning doesn’t hurt and makes the project go a lot smoother.

4.3.2 Recommendations to Course Administrators

• Use the same project rooms.
  The project rooms used in the project were wonderful. Large rooms, with great view over the school grounds were superb!

• Consider shortening the Scrum course and prolonging the Erlang course.
  Unfortunately we feel that you might need more time on the Erlang course. The tight schedule made it hard to follow the course content. Since the Scrum course ended early and you probably could have shorten it more, a few more days of Erlang had been possible.

• More microwaves and coffee machines.
  One more coffee machine and microwave could had been useful under lunch hour and on coffee breaks. People often ate at different times, because there were long queues.
5 Conclusion

This project has overall been a positive experience for us all. We have got the opportunity to experience what real life projects can be like, how it is working in a large group, what problems follow and how to cope with them. We have experienced the stress of working under time pressure, obligations of being in time for work, and delivering a good product in the end.

Not only we had the opportunity to make use of all the knowledge we have gained from years of study in this field of work, but we have also gained a lot of information in other ways than being taught by a lecturer.

This project has not only given us knowledge in those technologies used and project methodology, we have also gained lifelong experience in cooperation with people and companies that will surely help us in future careers. Mistakes have been made, but in the end we have gained a lot of knowledge which we can add to our respective key-value store.
6 References and Websites


A Contributions

A.1 Christian Rennerskog
Throughout the project I’ve been seen in many parts of the development. In everything from Nitrogen pages to Riak APIs. In the beginning of the project I started out with developing the Nitrogen accelerated web pages in ways of creating structure of the pages. Most of this work was put into the administration pages in the beginning but changed to include most web pages later on. When investigating how we could expand the customization of the pages I initially took part in integrating the ErlyDtl template system. A lot of my work effort has been put in re-factoring the web pages since the main structure changed a lot during the project due to API changes and integration of ErlyDtl.

Later in the project I designed and implemented some of the APIs for the database, together with their respective web pages (e.g. roles and users). Throughout the project I solely took on the task of creating the web page designs.

I’ve also taken the role as SCRUM-master during two sprints of this project, in which I believe I have structured our way of using the methodology. In the end part of the project I’ve written almost the whole of this report (course report).

A.2 Manzoor Ahmad Mubashir
In this project, I was associated with the Nitrogen team which was working on the presentation layer, i.e. development of new nitrogen pages that also involved communication with the Business Layer (Database APIs). We had two sub teams in the Nitrogen team itself: one was working on the customer side and the other on the administration side. I was working on the administration side. However, when required, I also had the opportunity to work in other modules including making very few changes in the Database APIs. Moreover, there were instances when the underlying functionality had to be changed which led to changes in the Database APIs and the layout of the web page also. As a result, I had to make changes accordingly in the modules I was working on.

Moreover, I wish to mention two specific exciting tasks that I undertook. One was to learn and implement the email sending capability of Erlang and second was to investigate and implement the GEO-IP (finding geographical locations from the IP address) in Erlang. These two tasks were definitely exciting and challenging for me but after thorough research and effort, I was finally able to demonstrate their working in time. In short, this project has been a tremendous learning experience for me and I have benefited from it a lot.

A.3 Nicklas Nordenmark
Early on in the project we split up in two distinct developing teams, one that developed Nitrogen accelerated web pages and one that focused on the develop-
ment of communication with the key value store of choice, Riak. I started out doing mostly database related tasks, such as developing some of the early APIs as well as figuring out the structure of the storage of information. Also had numerous discussions regarding fundamental things related to finding solutions to some of Riak's problems (indexing, listing etc.). Later on as most of the database structure as well as the APIs were completed I moved on to work a lot on the generic plug-in system which turned to be a huge task with many changes in design along the way.

When we started to investigate customization even further I put a lot of work into integrating the ErlyDTL template system. Along with the template system came the concept of global variables which also turned out to be a huge task that still would require a lot of work.

When we started out on the project we had no idea how any of the key concepts worked and I feel that I’ve learnt a lot of (especially) Erlang throughout the project. I’ve also realized a lot of things that could have been done differently from the start and I’m confident that building an E-commerce framework with the knowledge I’ve gained now would be a much easier task than what it was for starters.

### A.4 Kim Honkaniemi

Already from the start I knew I wanted to get more involved with the technical parts of this project, rather than the designing of the web-pages or overall system design. So I started by digging into RIAK and soon after that started to design and program our first database APIs. Along the way I’ve come to learn how important smaller things are, for example coding standards, when several programmers work on the same type of modules, for example the APIs. When the APIs starting to get ready and only minor bugs and smaller tasks where left to do in them, I moved over to start investigating and create things like the search-system, interval filtering for database fetching and the system for distributing content over all the Nitrogen nodes, also known as N'Sync. I did try some smaller work in Nitrogen and system overall design, the bigger contribution of these would be the admin page for showing status about the Nitrogen Cluster. There have also been smaller contribution by helping others by solving a lot of problems and suggesting solutions.

I’ve gained a lot of experience regarding how to work in group, how to distribute work and obviously the improvement of me as a programmer. I did sign up for interest to act as SCRUM-Master, but after watching other people as SCRUM master I’ve learned the things about the SCRUM-Master role I wanted to learn, so I felt that trying it out wasn’t something I would wanted to do at this point.

### A.5 Tanvir Ahmad

In this project I worked with the Nitrogen framework team. At beginning we divide our team into parts one team for administrator website and one for
client website. I worked on client web pages. I created new Nitrogen pages for listing categories and products. I developed complete cart of this project. I also worked on different layers of this project for example cookie_cart.erl, utils.erl etc. I also integrate the new Django Plugin system into my pages it was really a fun when I divide my HTML pages into layers. I also got a chance to work on user Registration and user login system at customer web site. Regarding cart I have implemented different functionalities e.g client can save his carts, and when ever he wants he can check in for this saved cart, user can also view orders from client side after logged in. I also developed a mechanism for checkout that how anonymous user can buy products without log in from our shop. I also got a chance to work on the administration pages. Displaying orders, correcting user information on administrator pages.

A.6 Shahzad Gul

At first, I worked in the Riak team and involved in building structure of the initial database. After that, I was associated with the Nitrogen team throughout the whole project. I was also the scrum master for one sprint.

My core work on Nitrogen was to design the administrative configuration logic and pages along with their related APIs. I was also responsible for the internationalization of pages. I have written the DB and wrapper API's for easier implementation of internationalization on both the admin and client side. Moreover, I worked on few admin and client pages. One of the splendid experience was to become the scrum master for one sprint. It was a good experience to lead the team as a facilitator.

In nutshell, It was a great opportunity to be involved in hands-on experience and I tried my best to avail this chance. As a programmer, I have gained plenty of experience. Furthermore, I enjoyed the collaboration and cooperation among the team throughout the project. At end, I wish to all members of project best of luck for their future!

A.7 Yujuan Zou

At the beginning of this project, the team was divided into two groups, each has one focus. Since I am an usability specialist in this team, I decided to join the Nitrogen group, which focus on the interface design and development. This project is to develop a e-commerce web framework, which I did not have much experience before, so I started with reviewing articles and website about usability of e-commerce system. According to the characteristic of this website, I have filtered and listed a number of usability-related guidelines that would be important in developing user-friendly interface, which need to be paid intention, especially at the early stage of development process. I have hand over the guidelines to the whole development team and let them read it especially the people developing the Nitrogen interface. After that I have made the site-map of the web framework based on reviewing similar e-commerce website and project requirements. Then I have worked together with one interface developer on
the administration site which has more usability issues involved, and at the same time, I also learned a lot about the whole web structure and some Erlang programming skills. Throughout this project, I have continuously evaluating the website based on the usability issues and prioritize the work based on time and resource availability, there is always no fully completed work, and I have attached a number of usability issues that can be improved in the report based on the final heuristic evaluation.

This project is really a big challenge for me, since I do not have much previous programming skills, and no idea about functional programming. I also do not have any experience on developing e-commerce website, I need to do a lot of research on it. I also need to communicate efficiently with my team member and convince them about the importance of website usability and its effects on e-commerce system. At the end of this project, I gained a lot experience about it and now I am more confident with developing user-friendly e-commerce websites.

A.8 Daniel Widgren

During the project I first worked with designing on how we should use Riak, this was very different from designing how a relation database work. After that I started to work on getting Edoc working and get the group to document their code. Worked on implementing various function in both nitrogen web pages and API for Riak. I started with Riak but later in the project when there was more things needed in Nitrogen I helped out there. During my short time as server administrator I did a workstation auto-install script that helped to reinstall all programs that we need to work on the project.

With background in economic I helped with the economic part of the e-commerce framework. I was involved in designing how orders, products and tax should work together. This was a challenge to try to get a good system that was easy to get a good first view but also helped the owner to get the information that was needed.

I also was group leader for the project group, and hold three presentations for the course. One was for the new students in DV/IT, second for institute and third for open public. Also helped the new students during interviews to see what we were working with. This was very good training in how to meet with people from companies but also trying to do Pr and spread information to the group.

A.9 Michael Nordström

I was the scrum master for the team’s very first sprint. It was my suggestion that we split into two research teams, one focusing on Nitrogen and one on Riak. This division, while not intended, stuck with us throughout the project. During the first couple of weeks of the project I took on the unofficial role of server administrator, installing and configuring all subsystems used by cookie cart (Mochiweb, Nitrogen, Riak) and also various development tools (Subversion, Hudson, Redmine).
As for development I was heavily involved in the design of both the template and plug-in systems. Once we got the first version of those systems up and running I made the first prototype of our built-in editor. Being one of those who had the most knowledge of the plug-in system I also developed our payment system. The payment system is entirely based around plug-ins, where each payment option is its own plug-in. To communicate with Klarna I translated an provided Java based xml-rpc API to an Erlang based equivalent. The translation itself was quite easy, but fixing the few bugs that occurred proved quite challenging. When everyone else was away during the holiday break I took the opportunity to overhaul both the template and the plug-in system. Since most of our system is built on top of these taking them down and rebuilding them during a normal sprint would have crippled the teams progress during that sprint.

Furthermore I was assisting group leader and, together with Daniel, worked on three presentations during the course. The first was for the new DV/IT students, the second was for the IT department and related companies, and the third was for the open public. Along with Daniel we were contact persons for new students who were supposed to conduct interviews with us for a report they were supposed to write.