BLEKINGE TEKNISKA HÖGSKOLA
KARLSKRONA, SWEDEN
April 27, 2015

ET1446: SOFTWARE DEVELOPMENT FOR TELECOMMUNICATION SYSTEMS

This document is a superset of the Project Proposal document in the context of building an enterprise chat system for XtremeCompany.

PROJECT SPECIFICATION V1.0

Group: SWAT Kats

Sokratis Papadopoulos	Jibraan Singh Chahal
Johannes Grohmann	Jyoti
David Alarcón Prada	Akanksha Gupta
Gautam Vij	Rohit Raghav
Inanc Gurkan	

Publication Date: April 27, 2015 **Version Number: V1.0**

PROJECT SPECIFICATION

1. PREFACE

This document contains a project specification to create a chat client called "**LetsTalk**" for XtremeSecurity. The document is specifying important aspects which are to be executed by our team "**SWAT Kats**".

In the following we will first provide a glossary with a list of abbreviation in chapter 2. After that, the environment as well as the problem are described in section 3. Section 4 contains our solution for the described problem, while section 5 states some limitations that our product will not include. In addition, a rough time plan is given in section 6 and a project organization in section 7. Then it is configuration management at section 8 and progress tracking in section 9, while quality control is presented in section 10 and risk management in section 11. Finally we present the system release plan and we refer to the references.

VERSION HISTORY OF THIS DOCUMENT:

Release v1.0 on 2015-04-27

- Initial release

2. GLOSSARY AND ABBREVIATIONS

In this section are defined technical terms that are in this document in alphabetical order.

- **Dev Team**: Developer Team
- **e.g**. : exempli gratia – for example
- **etc**. : et cetera
- **GUI**: Graphical User Interface
- **IP**: Internet Protocol.
- **Metadata**: information about a certain item's content
- **SQL**: Structured Query Language.
- **WBS**: Work Breakdown Structure.
- **Bin**: binaries
- **Src**: source
- **Lib**: libraries
- **Docs**: documents
- **PDF**: Portable Document Format
- **HTML**: Hyper Text Markup Language

3. BACKGROUND

We are required to design a communicating tool among the employees of the company XtremeSecurity and satisfy the specific needs of its travelling salesmen.

The employees will be able to communicate over an encrypted network and share binary files as needed. The main purpose is to provide our clients with the means of fast, efficient and reliable communication over the distance with the goal to help them increase their time utility.

The salesmen often require access to the latest marketing materials, white papers, company brochures and user manuals in order to showcase the products to potential customers. Proper update of new plans in the respective sector of sales is the top priority for them to present best plan to their clients. The work office of XtremeSecurity is situated in Stockholm and are supposed to travel to different destinations to meet with their clients. They are required to convince their clients over some package and may need some details from their center office. Support Engineers out on the field may also need more data from their headquarters to show it to their clients. Our product "**LetsTalk**" will resolve the issue of carrying around the updated soft copy of various packages to be shown to their clients.

4. PROPOSED SOLUTION

The project aims to provide the customer with a secure fully functional communication system. The customer for the project is a company, which wants its employees to properly communicate and work together. We will provide all the various collaboration tools that allow its employees to have secure communication from different geographic locations.

The communication involves text messages and exchange binary files of arbitrary size to access certain documents such as the latest marketing materials, white papers, company brochures and user manuals in order to showcase the products to potential customers. The customer also needs on field engineers to be able to securely download patches and service packs to resolve customer issues on location.

A central database is required to store user information metadata associated with each sent or received message, so that every user can see the date and time a message was sent, the name of the sender and recipient, if the message was read by the recipient and when it was read and can browse the history of their conversations with various peers. To provide fast service the messages should be sent to offline users as e-mails to reduce the delay. User can see a message status (read/unread, message time stamp). A call record, timestamped information about the sender and receiver of messages, their corresponding IP addresses, along with metadata such as message size, type (message or file) should be kept in a SQL database and only authorized users (administrators) should be able to access it. The customer requirement also includes an administrative role: delete a user, block and unblock a user, emergency broadcast message like some crash problem. The administrator can view statistics of records graphically as tables/graphs (GUI) available on main database server (SQL).

The project will establish a secure chat for authorized employees .The administrative access will be granted by authentication based on username/e-mail address and password. To ensure the security of confidential client data the chat messages will be encrypted. File exchange with live-status (used/remaining time for upload/download) will be provided. A user can temporarily block a user from the address book and an address book for the users will be displayed with associated status (offline, available, idle, busy: available & busy are handled manually by user, whereas offline & idle by the system).

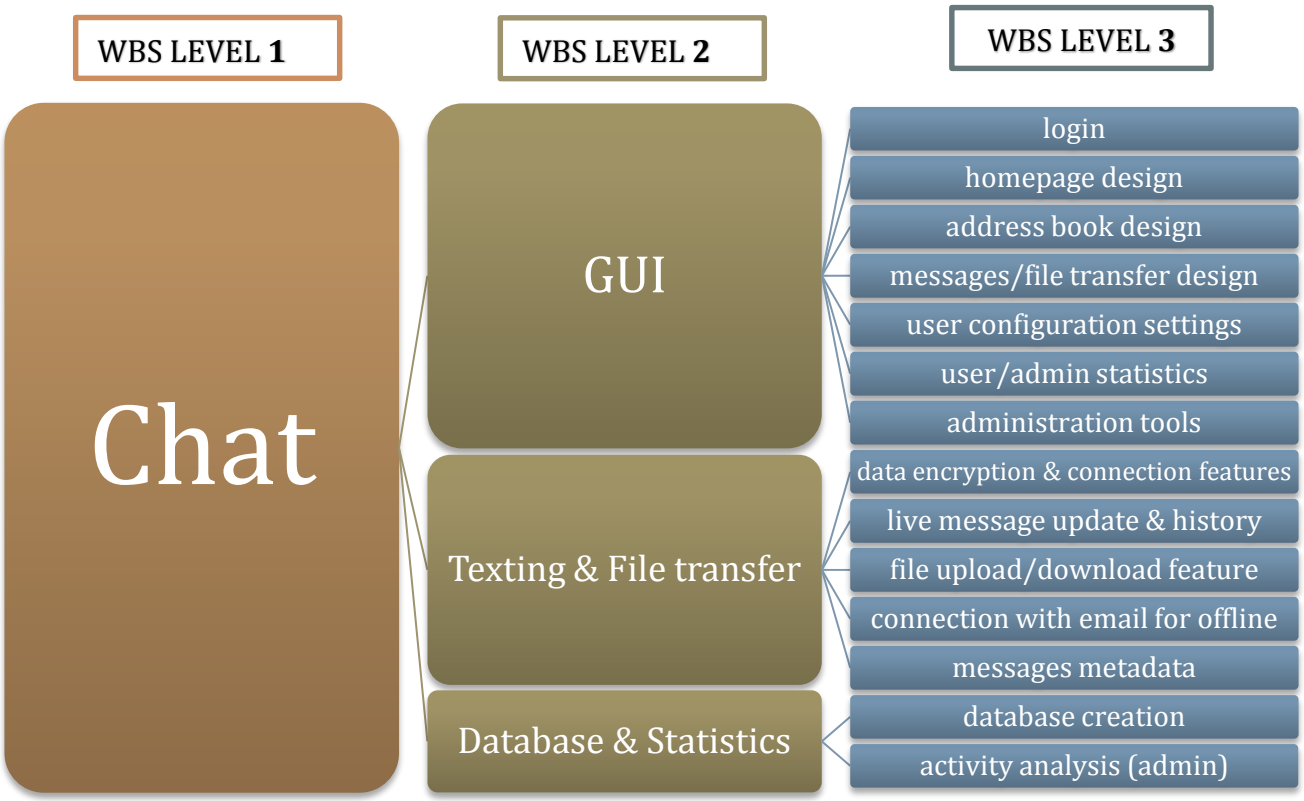
5. LIMITATIONS

Although the chat messages are sent instantaneously, this is no real-time client. Since messages and files are usually transmitted over the internet, delays of various lengths can occur. Furthermore in extreme cases (e.g. internet connection lost), messages or files can also be lost, transmission is never guaranteed and transmission time may vary.

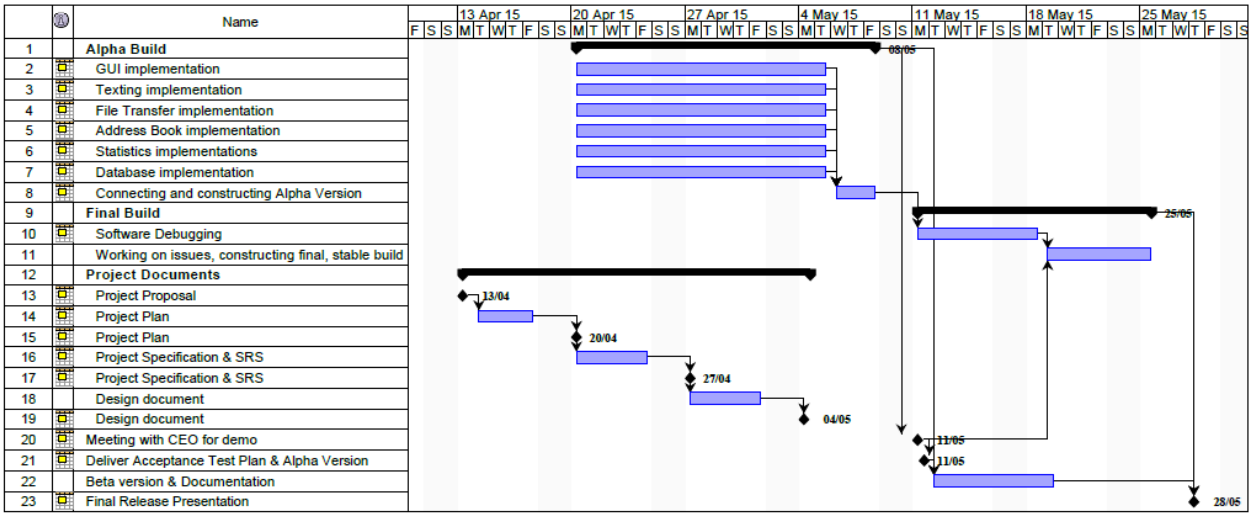
The client is not supposed to support any acoustic support or emoticons in the chat. Only private chats, are supported (no group-chats or broadcast options).

6. TIME PLAN

After describing the background, our proposed solution and its limitations it is time to present the time plan of the project. We have developed a WBS structure that shows the whole picture of the project, broken into small parts. Also, we have written down all the milestones and tollgates that this project will follow within the time budget.



MILESTONE (blue) / TOLLGATE (green)	Deliverable	Receiver	Time
Deliver initial project proposal	Project Proposal	CEO	**13 April**
Deliver updated project proposal & project plan	Project Proposal, Project Plan	CEO	**20 April**
Start line of Software development	Split of work	Dev Team	**20 April**
Deliver project specification & SRS	Project Specification & SRS	CEO/Customer	**27 April**
Deliver design document	Design Document	CEO/Customer	**4 May**
GUI implementation	GUI of Software	Dev Team	20 April - 5 May
Texting implementation	Texting of Software	Dev Team	20 April - 5 May
File Transfer implementation	File Transfer of Software	Dev Team	20 April - 5 May
Address Book implementation	Address Book of Software	Dev Team	20 April - 5 May
Statistics Implementation	Statistics of Software	Dev Team	20 April - 5 May
Database Implementation	Database of Software	Dev Team	20 April - 5 May
Connecting and constructing Alpha Version	Package of Alpha Version	Dev Team	**7 May**
Meeting with CEO for product demo, providing basic functionality of alpha version	Software	CEO	**10 May**
Deliver Acceptance Test Plan and the Alpha Version of the software	Alpha version & instructions, Acceptance Test Plan	CEO/Customer	**11 May**
Software debugging (each part is being debugged by the people responsible)	Debugging results	Dev Team, CEO/Customer	**18 May**
Beta version of our software accompanied with a full documentation	Beta version & Full documentation	CEO/Customer	**18 May**
Working on issues, constructing a stable final release	Stable software	Dev Team	**25 May**
Final Release of Chat → Final version of the program. Show to CEO/Customer. Explain the support provided.	**Final Release**	CEO/Customer	**28-May**



7. PROJECT ORGANIZATION

GUI LAYOUT	
Login interface	Sokratis Papadopoulos
Basic GUI for the messages, user/admin statistics, admin tools	David Alarcon Prada
GUI for other users' status in address book, the user account configuration (change password etc.)	Sokratis Papadopoulos
GUI for address book, user status, file upload/download	David Alarcon Prada
TEXTING / FILE TRANSFER	
Data encryption & Connection Features	Johannes Grohmann
Live message update and Message History	Akanksha Gupta
File upload/download feature	Jyoti
Connecting the email to chat (for offline)	Johannes Grohmann
Developing the message status function	Akanksha Gupta
DATABASE AND STATISTICS	
Login tables and queries	Jibraan Chahal
User account configuration tables and queries	Rohit Raghav
Group Chat tables and queries	Gautam Vij
Users activity analysis, admin features	Inanc Gurkan
Single Chat tables and queries	Jibraan Chahal
File transfer tables and queries	Gautam Vij
Address book tables and queries	Rohit Raghav
Block list queries	Inanc Gurkan

8. CONFIGURATION MANAGEMENT

8.1 VERSION MANAGEMENT

A version control system is a repository of files, often the files for the source code of computer programs, with monitored access. There are a lot of possibilities for new feature upgrades in future in different components of the system. Since we have different members of our team working on different components, it is very important to record the changes and build operations done by different developers from our team. Git system, a distributed revision control system is used as a support for distributed, non-linear workflows. This system is used as a part of learning phase for this course and to record all the changes made in the development process to ensure smooth working and avoid any interference between each other.

8.2 SYSTEM BUILDING

A method of building in which prefabricated components are used to speed the construction of buildings. Main focus would be put in building a decentralized structure of different functionalities that are to be implemented. Smart build operations will be used to for minimal compilation using "make" operations. The application will be developed in Java. At the end of the development of application, "Junit" will be used to run automation test and to confirm the working of all the functionalities. All the individual XML files generated by the JUnit task will be used to provide a browsable report of the test cases results. We have used Microsoft Word for document generation.

8.3 RELEASE MANAGEMENT

Our application will have a product release on 28th may and then in future few minor releases depending upon the feedback and the bug reports. All libraries and the compilers used will be properly documented along with the build information and other tools configuration used. One executable file or an installer file will be provided during the release of the application.

9. PROGRESS TRACKING

Due to the lack of a project manager, all progresses of all tasks have to be supervised by all team-members themselves. Since every task has a responsible team member, this member is also responsible for the *delivery of the task on time.*

Communication is crucial on this topic. It is very important to communicate upcoming delays or resource contentions with all relevant team members, in order to be able to react properly.

The weekly meetings perform a key role here. In this meetings, every team member has to report the progress of his/her assigned task to the plenum. Any problems and/or delays shall be reported and discussed by the team. If necessary additional resources can be assigned to the task, in order to finish in time. Although it is in the responsibility of the assigned team-member to report delays, problems or resource contention, it is the authority of the whole team to deal with upcoming problems.

Delayed or missed deadlines shall also be included in the Project-Libre-file in order to communicate them to all team members, as well as the CEO in severe cases.

10. QUALITY CONTROL

The successful outcome of virtual anything requires good, up-front planning. And quality is no exception. Controlling the quality of product throughout development is imperative to ensuring that the end deliverable is going to be at a level that maximizes the enhancements and minimizes any issues. Following we explain how is this going to be the case in our program.

Essential is the communication not only among the team but also the establishment of a great connection with the CEO & customer. Having a constant communication with them will lead us to useful feedback that will perfectly adjust the product into customer needs. So, we value communication as a great principle for achieving the best possible quality of our product and it will be the best way to build exactly the product that customer wants. Alpha and Beta versions of our product will boost their feedback and will lead us to develop exactly the product they demand.

As for internal work, as developing the software, weekly meetings can lead the control checking. Each member has been assigned with a part of the project and he will inform and present to the rest of the group for his progress on it. That way, everyone can check, feedback and suggest ways to overcome problems that will may occur. Plus, we will be constantly checking if whatever was promised in proposal will be met on the final release, point by point.

Regarding software quality control, we are planning to use JUnit that will support us in accomplishing a great testing period for our software. In that period, we will also be able to clear out the code, break down longer functions, making it easily understandable by everyone.

11. RISK MANAGEMENT

CHANGE OF TECHNOLOGIES

After an appropriate project proposal and workload assigned; technologies to be used were selected. Change of technology risk can be human-sourced and software-sourced. The technology that being used may not satisfy developers' project plan. Disadvantages of these technologies are subject to change. As for the software factor; the technology with changes may affect the software. Customers are supposed to use the software in their PCs without any compatibility problem.

Detecting human-sourced is tough. The technologies used were selected by the same Dev Team. In other words; if the Dev Team needs a change in technologies, that is the result of bad experience with the technology right in this project. The technology that is subjected to change may/will affect the project depending on how far is the project developed, and the number of dependencies used the technology yet. This is a risk that affects project crucially. Team SWAT-KATS are prepared for this risk. Technologies were studied well before the start of development and some technologies changed (such as FTP). Brainstorming sessions and team meetings were handy lowering the impact and the probability of happening of this risk. The risk will have an impact on project as the project is developed in parallel, each part is dependent to one another; yet SWAT-KATS have managed to lower the probability.

Software-sourced risks are not meant to happen in this project's timeline. By software-sourced risks, it means; changes in a technology that is being used. Even "improvements" might affect the project in a bad way. As the project is planned to finish around 2 months, no crucial changes expected in the technologies that are used. Software-sourced risks have severe effect in this project as it can be fixed quickly. The possibility of happening is low. So, this risks is not considered as a "dangerous" risk nor "likely to happen".

CHANGE OF CUSTOMER DESIRES

Software projects are subject to change in anytime. A good software is improvable. The project is being planned and developed with fully understanding of that statements; yet, change of customer's desires stands as a problem. These changes affects project plan and the project structure. Analysing customer's needs is crucial for avoiding this risk. It is hard to foresee the impact of this risk to the project but it can be stated that it will affect the project. Misunderstanding and/or miscommunication between client and the team leads to this problem. SWAT-KATS analysed the user needs and proposed improvements -as additional functions. The project structure is rigid at this moment. All pros and cons and limits of technologies were discussed, and then comes the rigid structure.

This risk is not likely to happen, and if it happens; the impact won't be crucial because of the maturity of the structure. By "rigid", it doesn't mean it is not "changeable" but many scenarios have been thought of. Thus, this risk is not considered as a "dangerous" risk.

DATABASE CHANGES

The project is developed in parallel, following the work plan with respect to the workload assigned. Database has a crucial part. Database needs to represent all information in an efficient way. A not well-structured database still can do its part, but difficulties will occur in all other areas as database is dependent to every other area, vice versa. Changes in database will affect whole project in different ways. It may improve or worsen the differents areas, but it is easily seen that; changes in database will lead to changes in project which means it will cost time. Its cost is only time in a good scenario, on the other hand; it may have a huge impact on project.

This risk is being handled with brainstorming sessions and team meetings where every member can state their thoughts. The database structure is decided and it is not likely to change as it contains point of view of every member. Database changes may or may not be crucial, but we have trust in the project structure. So, this is not considered as "dangerous".

SECURITY OF MESSAGES

Messages are sent securely and encrypted through server. The server stores every message with information of sender, receiver, time etc. If an attacker manages to penetrate the server security, can reach the messages, unencrypted. Although this is a scenario that is not likely to happen, it has a huge impact. The messages being sent is decrypted in server, and encrypted for the receiver. If the message is stored encrypted in the server, it uses more resources. As this risk is not likely to happen and the

solution may damage system more than the risk itself; no action will be taken against this risk as a priority.

USER INFORMATION SECURITY

SWAT-Kats takes security seriously. The processes are well encrypted, and user passwords are stored hashed in server. Possessing a user name may leads to the possession of the password. In an attack to the server; we don't want the attacker to access user names -passwords are secure as they are stored hashed. Hashing user name can be a solution to this risk. Not only passwords, but also user names; so that we can provide more safety. By hashing user information; the message security problem's impact is lowered. The messages would have the hashed information of users. That approach helps protecting security along with all of the precautions.

The solution to this risk is being considered now. Solving may cost more than the actual impact of the risk. Another solution is being thought of. The current solution is easy to implement, so the risk is not considered "threatening" and "likely to happen" yet still it is worth to discuss.

AGENDA

The time planning and scheduling have already been done. But in a project like this; where all of the development areas are connected tightly to each other, a simple bug or delay usually causes bigger delays that makes hard to follow the planned schedule. Facing this risk since the start of the project, our planning skills as a team is improved. This risk was highly likely to happen. As a solution, schedule and current state of development is reviewed and updated (if needed). So, this risk doesn’t have an impact on the project and the probability of happening lowered. We, as SWAT-Kats, are trying our best to keep it that way; keep the project following the plan

TRANSMISSION IS NOT GUARANTEED

This is a risk is stated as a “limitation”. The meaning of this risk is; messages can be lost. To make the message transfer efficient, and because solving this risk may cause the worsening of the project; it is decided to stay as it is. This risk is highly unlikely to happen and the fix of this problem may need a structural change. That is why this risk is not prioritized.

12. SYSTEM RELEASE PLAN

12.1 TESTING PLAN

Requirement	Time	Test
Server Connection	2nd May 2015	Access to information on central server
User Login	4th May 2015	User should be on home page after login.
Text Messages Exchange	7th May 2015	Messages should appear on both users' screen.
Address Book Update	10th May 2015	If a contact is added, then it should appear in user's address book and they should be able to chat with each other. If a contact is blocked then it should not appear in address book and they should not be able to chat with each other.
Files Exchange	10th May 2015	Users must be allowed to exchange files with each other and the GUI must show statistics on the used and remaining time for upload/download and a time estimate for completion.
Encryption	14th May 2015	Encrypted password , messages should be saved to the database
Online Status	16th May 2015	Status of user should be changed and other contacts should see the same status which he sets.
Offline Chat	20th May 2015	Test in a scenario with the unread message and offline user to check the email notification service.

12.2 PACKAGING PLAN

We will provide two .tar packages. One for the server to be distributed to the administrator and the other will be for the clients. Each package can be extracted in the working directory. Each package will have an executable .jar file. Along with it our package will have different sections- bin, lib, src, doc with data as an additional in server package.

- **Bin**: It will contain all the binary executables of the source files.
- **Data** (server only): This will contain the .db files for the database used by the clients for old chat messages, files, address book etc.
- **Lib**: It will contain all the java libraries used up in all the source files.
- **Src**: It will comprise of all the java source files along with image or audio files for GUI.
- **Docs**: In the docs/ directory we have the documents that we produced in our project (e.g., the Project Libre plan, the SRS, the design doc, the acceptance test plan).

We will submit the final release by itslearning & git server on **18th of May**.

12.3 DOCUMENTATION PLAN

12.3.1 Installation documentation

We will provide a PDF document where the software responsible of XtremeSecurity can find information about how install & configure the program in each employees' computers.

Because we are going to provide the program through .jar file in order to let execute the software in all Operative Systems as be possible, we will explain how to install the JVM corresponding to run the

program. All explanations will be written with clear sentences and will be attached with useful images.

12.3.2 User documentation

Explain what sort of user documentation you will provide (HTML, LaTeX, PDF etc.) and what it will cover. Provide a time schedule.

We will make a PDF manual where users and system administrators can find information about how use the program.

It will consist of an easy guide, which will contain screenshots about different options available clicking different buttons. Thereby users and systems administrators will be able to learn to use the tool by themselves.

12.3.3 Developer documentation

What documentation will the developer obtain (useful when a framework, API or function library is developed). Provide a time schedule.

The main code with english comments in all methods in order to understand the code and to develop future extensions. Then we will generate a Javadoc document, and as developer documentation we will include all the documents delivered in checkpoints.

A PDF will be provided with the full explanation of the working and structure of the Database. It will contain the graphical representation of the tables and their connections with each other.

12.3.4 Documentation time schedule

MILESTONE (blue) / TOLLGATE (green)	Deliverable	Date	Dependent Upon
Alpha version documentation	Installation & User documentation	11 May (Checkpoint 4)	Alpha version of the code
Beta version documentation	Installation, User & Developer documentation	18 May (Checkpoint 5)	Beta version of the code
Final release documentation	All documentation complete and ready to ship	28 May (Final Checkpoint)	Documentation complete. Approved for release.

13. REFERENCES

- http://oss-watch.ac.uk/resources/versioncontrol
- https://ant.apache.org/manual/Tasks/junitreport.html
- http://www.istc.org.uk/wp-content/uploads/2011/08/Beth-DOCPLN_TPL.doc