[CS3984 Cloud Software Development](http://manta.cs.vt.edu/cs3984/)

Spring 2017

**SaveLives**

Firstname M. Lastname, [emailAddress@vt.edu](mailto:emailAddress@vt.edu)

Firstname M. Lastname, [emailAddress@vt.edu](mailto:emailAddress@vt.edu)

Firstname M. Lastname, [emailAddress@vt.edu](mailto:emailAddress@vt.edu)

Firstname M. Lastname, [emailAddress@vt.edu](mailto:emailAddress@vt.edu)

Firstname M. Lastname, [emailAddress@vt.edu](mailto:emailAddress@vt.edu)

Firstname M. Lastname, [emailAddress@vt.edu](mailto:emailAddress@vt.edu)

Firstname M. Lastname, [emailAddress@vt.edu](mailto:emailAddress@vt.edu)

Department of Computer Science

Virginia Tech

Blacksburg, VA 24061

*Date:* January 17, 2017

*Team Number:* 1

*Instructor:* Prof. Osman Balci

*Supervisor:* Dr. Laurian C. Vega  
 [Laurian.Vega@nextcentury.com](mailto:Laurian.Vega@nextcentury.com)

EXECUTIVE SUMMARY

TBD

(Provide a structured summary of the content of the document here, by emphasizing your important contributions to catch the attention of a very busy executive.

\*\*\*\* **IMPORTANT**: Provide here a numbered list of cloud software features and other technology features you implemented in your cloud software application delivered. \*\*\*\*)

**TABLE OF CONTENTS**

EXECUTIVE SUMMARY ii

1. SOFTWARE LIFE CYCLE 1

2. PROBLEM SPECIFICATION 2

2.1 Description of the Problem Context 2

2.2 Cloud Software Features to be Implemented 3

2.3 Database Management System 4

2.4 Complexity 4

3. REQUIREMENTS SPECIFICATION 5

3.1 Functional Requirements 5

3.2 Non-Functional Requirements 6

4. ARCHITECTURE SPECIFICATION 6

5. DESIGN SPECIFICATION 6

6. DELIVERED SOFTWARE FUNCTIONALITY 7

7. STEP-BY-STEP INSTRUCTIONS FOR DEVELOPING NEW FEATURES 7

7.1 New Feature 1 Title 7

7.2 New Feature 2 Title 7

7.3 New Feature 3 Title 8

8. SUBMISSION INSTRUCTIONS 9

9. CONCLUSIONS 10

10. PERCENTAGES OF CONTRIBUTION 10

REFERENCES 10

Calculation of Grades based on Percentages of Contribution 11

Team Project Requirements 11

Grading Sheet 12

# SOFTWARE LIFE CYCLE

A **good software engineer** develops software by following the software life cycle shown below.



A **programmer (hacker or ad-hoc developer)** develops software by looking at the problem and directly coding in an IDE. This approach is known as the **Build-and-Fix Approach**, which must never be used!

# PROBLEM SPECIFICATION

This section presents a description of the problem to be solved by way of engineering a cloud software application.

## Description of the Problem Context

Wikipedia reports that “The city of Baltimore, Maryland, United States is infamous for its very high crime rate, including a violent crime rate that ranks high above the national average.” In an article entitled “After Bloodiest Year, Baltimore Murder Rate Still Rising”, Huston indicates that “With 344 murders, last year was the bloodiest year in the City of Baltimore’s history. But as 2016 reaches its halfway point, the bloodshed seems to be showing no signs of stopping.” [Huston 2016].

The law enforcement officers need help to prevent crimes. One approach to help the Baltimore City Police Department to prevent crimes is to provide them with crime data analytics and mapping so that they are better prepared about what, where, and when to expect occurrences of illegal activities and crimes. In an article entitled “Predicting Crime Using Analytics and Big Data”, Collins indicates that “A unique approach to crime analysis may allow police officers to predict illegal activity.” [Collins 2013]

The Baltimore City Police Department already provides victim-based crime data at <https://data.baltimorecity.gov/Public-Safety/BPD-Part-1-Victim-Based-Crime-Data/wsfq-mvij> on their Open Baltimore website. This data can be ingested into persistent storage using **MongoDB** database [ <https://www.mongodb.com/> ]. MongoDB is a free and open-source cross-platform document-oriented NoSQL database management system. Once the data is stored in a MongoDB, it can be injested using RESTful API called SODA (Socrata Open Data API) documented at <https://dev.socrata.com/foundry/data.baltimorecity.gov/4ih5-d5d5>.

As a solution to the stated problem, the Baltimore City Police Department would like to have a multi-tiered distributed cloud software system to be able to track crimes in order to support increased police support during peak crimes and locations as needed. By looking at trends and anomalies, the law enforcement officers can better understand the nuances of crime and better support the city.

The desired cloud software system can be reused by many stakeholders including other city police departments, FBI, and the Department of Homeland Security by using other collected crime datasets.

## Cloud Software Features to be Implemented

List the features you will implement, such as:

|  |  |
| --- | --- |
| 1 | Activity data CRUD operations |
| 2 | An upvote/downvote system to allow users to cast their vote on posted decisions |
| 3 | Email (Email-based notification system to allow users to set reminders based on work items) |
| 4 | Email (Email-based notification system to notify when a user takes an action) |
| 5 | Email (Feedback Email sender via JavaMail API) |
| 6 | Email (Java Mail API , Automated Email Service) |
| 7 | Extract Geolocation data from an uploaded photo and show its location on a map |
| 8 | File Upload (photo file) with CRUD operations |
| 9 | File Upload and Download |
| 10 | File Upload and Download (for documents, videos, images, BLOBs) |
| 11 | Google Maps with activity locations pinned |
| 12 | Java API for JSON processing |
| 13 | JSON Processing (JSON-P) |
| 14 | Member-Only Access controlled by the gatekeeper EJB component. |
| 15 | Newsfeed post and presentation via Database storage and retrieval |
| 16 | Photo gallery of users subscribed to an activity |
| 17 | Primefaces (Datascroller, HTML Editor, Pie Chart) |
| 18 | Primefaces (HTML Editor) |
| 19 | RESTFul Web Services |
| 20 | RESTFul Web Services connected to a mobile app running on an iOS or Android device |
| 21 | RSS Feeds |
| 22 | Timer Service API |
| 23 | Timing Services for managing the Newsfeed table in the database |
| 24 | User Account CRUD (User Database) |
| 25 | User Session Inactivity Timeout |
| 26 | View uploaded photos under different categories |
| 27 | Weather and Google Maps Widgets/API |
| 28 | WebSockets (chatroom) |
| 29 | WebSockets API |

1. **Requirement:** We develop **cloud software**, not a website! Therefore, the emphasis of your application must be placed on the server side, application logic, and storage and retrieval of data.
2. **Requirement:** You are **required** to use **PrimeFaces** to easily develop the user interface.
3. **Requirement:** Use of a **MySQL** database with JPA or **MongoDB** is **required**!

## Database Management System

Describe how the database (**required**) will be used for storage and retrieval in your application.

## Complexity

Substantiate that your cloud software application, to be developed by your team of   
*Solution Providers* — *Cloud Software Engineers*, will be much more complex than the tutorials covered in the course to justify 40% of the course grade.

# REQUIREMENTS SPECIFICATION

This section specifies the Functional and Non-Functional Requirements under which our cloud software application will be developed.

## Functional Requirements

**Filtering**

1. The system shall have interactive visualizations that filter the data displayed on the page. These filters will be for date range, crime type, weapon, and location. All filters will change what is on the page in under 5 seconds.
2. The user shall see a timeline of the data that embeds a barchart of crime occurrences. This timeline shall persist on the page regardless of the other visualizations.
3. Filtering or selection in one visualization shall affect the other visualizations
4. Users shall be able to view crimes by day and by hour to view temporal trends in the data.
5. Users shall be able to view the data by neighborhood within two clicks from any visualization (i.e., map, charts, table).
6. Clusters of crime occurrences shall be visualized within any visualization (i.e., map, charts, table).

**Charts**

1. Users shall have charts of crime by filter type. These charts will change depending on the selected filters.

**Map**

1. The system shall visualize the data on a map.
2. When the user clicks on a pin upon the map they shall be able to see the record details.
3. Map shall provide a heat map mode.
4. Map shall display features from a feature server.
5. When the application starts the data shall be shown on the map. This shall be rendered in the UI in under 15 seconds upon starting the application.

**Preferences**

1. Users shall be able to save and name their searches. Saved searches will persist across session.

**Table**

1. Users shall be able to see all of the data in a table view.
2. Users shall be able to search, sort, and filter on the table view.

**Other**

1. Before entering the tool all the data for the last 90 days shall be pre-loaded.
2. The application shall function in a modern browser.
3. All visualizations shall be 508 compliant.
4. The system shall be able to handle up to 15 concurrent users.
5. Data shall be ingested from the crime database daily.

**Stretch requirements**

1. Peaks or dips in the volume of crime occurrences shall be visualized within the UI.

## Non-Functional Requirements

1. The cloud software system shall be developed and delivered with royalty-free software products such as NetBeans IDE, GlassFish Application Server, MongoDB, and CentOS Linux.
2. The cloud software system shall be usable under the Chrome, Firefox, Safari, and Internet Explorer web browsers on any network connected desktop and laptop computers.
3. The user interface shall be developed in compliance with the best practices of usability engineering and UX.

# ARCHITECTURE SPECIFICATION

TBD

Specify the Java EE technologies used under the Client-Server Architecture together with the Service-Oriented Architecture if calling upon web services provided by other cloud software applications and APIs.

# DESIGN SPECIFICATION

TBD

Provide a graphical description (e.g., **storyboard**, **images, diagrams, drawings**, etc.) of the functionality of your cloud software application’s design.

(Set your file storage location folder name to FileStorageLocation-TeamN, where N is your team number.)

# DELIVERED SOFTWARE FUNCTIONALITY

TBD

Describe the functionality of your deployed cloud software application by using screenshots of user interfaces. Start with the URL for accessing your software. Then, screenshot by screenshot describe how your software is used. The entire software functionality must be described in sufficient detail.

# STEP-BY-STEP INSTRUCTIONS FOR DEVELOPING NEW FEATURES

TBD

Provide a list of the **new** cloud software features you have implemented in your project.   
A feature is a **new** feature if it is not covered in the course tutorials.

## New Feature 1 Title

TBD

* Describe the **new** cloud software feature you have implemented in one or more paragraphs. A feature is a **new** feature if it is not covered in the course tutorials.
* Provide step-by-step instructions with screenshots for how to develop the **new** cloud software feature, similar to the tutorials presented in class.
* Make sure that you provide references to the sources from where you have obtained the source code you used “as is” or with modifications. *Reuse of code is certainly expected and appreciated* as long as credit is given to the source.
* Make sure that your Java and XHTML code is very well documented.

## New Feature 2 Title

TBD

* Describe the **new** cloud software feature you have implemented in one or more paragraphs. A feature is a **new** feature if it is not covered in the course tutorials.
* Provide step-by-step instructions with screenshots for how to develop the **new** cloud software feature, similar to the tutorials presented in class.
* Make sure that you provide references to the sources from where you have obtained the source code you used “as is” or with modifications. *Reuse of code is certainly expected and appreciated* as long as credit is given to the source.
* Make sure that your Java and XHTML code is very well documented.

## New Feature 3 Title

TBD

* Describe the **new** cloud software feature you have implemented in one or more paragraphs. A feature is a **new** feature if it is not covered in the course tutorials.
* Provide step-by-step instructions with screenshots for how to develop the **new** cloud software feature, similar to the tutorials presented in class.
* Make sure that you provide references to the sources from where you have obtained the source code you used “as is” or with modifications. *Reuse of code is certainly expected and appreciated* as long as credit is given to the source.
* Make sure that your Java and XHTML code is very well documented.

Continue with the other new features, one subsection for each...

# SUBMISSION INSTRUCTIONS

Please carefully follow the instructions below in submitting your project:

1. Create a directory called lib in your NetBeans project folder. Place all of your JAR files in this lib directory. Right click the Libraries directory in NetBeans and select Add JAR/Folder... Import the JAR files into the Libraries from the lib directory by using *Relative Path*. This is critically important to run your project on another computer or location.
2. Set your file storage location folder name to FileStorageLocation-TeamN, where N is your team number.
3. Deploy your cloud software application to the server computer.
4. Using the deployed software, each team member must create an account with   
    *username* = last name (first letter capitalized), and   
    *password* = course website password.   
   Each team member must use the software application and create *meaningful content*, which is very important for evaluation and demo for your application.
5. Export (dump) your populated MySQL database content into a SQL script file using # mysqldump -u root -p NameOfDB > NameOfDBwithDBcontent.sql
   1. Move all DROP TABLE IF EXISTS 'TableName' statements in the exported SQL script file to the top of the file and *list them in a logical order* based on foreign key and delete cascading dependencies.
   2. *Reorder all of the table creations* with CREATE TABLE 'TableName' in a logical order.
   3. Correct problems with aposthrope as being the same as the single quote. MySQL exports character strings enclosed within single quotes. If there is a string like Dave’s it conflicts. So, if you have 'These are Dave's files.' change it to "These are Dave's files." to prevent single quote conflict.
   4. Test the corrected exported SQL script file on a computer and make sure that it can be imported to populate the database under the NetBeans IDE.
6. Create a ZIP file containing all of your project deliverables. Copy the ZIP file to a USB flash drive.
7. One or more team members go to Dr. Balci’s office to copy the ZIP file to his iMac, open the project under NetBeans IDE, create and populate the database, and make sure that the software is fully functional and running on his iMac. *The delivered source code and its documentation will be examined for grading*.

# CONCLUSIONS

TBD

# PERCENTAGES OF CONTRIBUTION

We hereby certify that the percentages of contribution specified below truly reflect the actual contributions of the team members. (Write your name as your signature)

|  |  |  |
| --- | --- | --- |
| *Student Name* | *% Contributed* | *Signature* |
| Student Name 1 | 20% |  |
| Student Name 2 | 20% |  |
| Student Name 3 | 20% |  |
| Student Name 4 | 20% |  |
| Student Name 5 | 20% |  |

The Virginia Tech Honor Code is fully in effect for the above declaration.

REFERENCES

TBD

Balci, O. (2017), “CS3984 Cloud Software Development Course Website,” <http://manta.cs.vt.edu/cs3984>

Collins, H. (2013), “Predicting Crime Using Analytics and Big Data,” May 24, <http://www.govtech.com/public-safety/Predicting-Crime-Using-Analytics-and-Big-Data.html>

GlassFish (2017), “GlassFish Application Server,” <https://glassfish.java.net/>

Huston, W.T. (2016), “After Bloodiest Year, Baltimore Murder Rate Still Rising,” June 6, <http://www.breitbart.com/big-government/2016/06/06/bloodiest-year-baltimore-murder-rate-rising/>

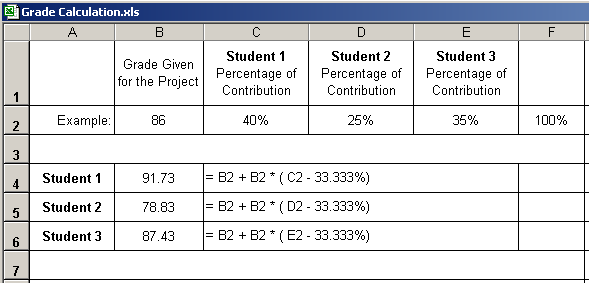
MySQL (2017), “MySQL Open Source Relational Database Management System,” <http://www.mysql.com/>

NetBeans (2017), “NetBeans IDE,” <https://netbeans.org/>

<< Include other references used in alphabetical order styled under References >>

\*\*\* Delete this and the following page in the final version of the report. \*\*\*

Calculation of Grades based on Percentages of Contribution



Additional percentage of contribution cannot exceed 10%.

If a situation arises where a student is doing more than 10% extra work, Dr. Balci must be informed immediately.

Team Project Requirements

1. Each team member is expected to contribute equally.
2. You shall submit percentages of contribution with signatures of all team members. In case of disagreement, you shall submit it separately with your rationale. (Write your name as representing your signature.)
3. Grades shall be determined based on the percentages of contribution.
4. If you contribute more than your equal share, then your grade shall be increased based on the extra percentage of contribution.
5. If you contribute less than your equal share, then your grade shall be decreased accordingly.
6. The extra percentage of contribution shall not be more than 10%. Dr. Balci shall be notified immediately if a situation arises where a student needs to contribute more than 10% extra.
7. A team member who does not cooperate with other team members for conducting the project with equal contribution shall be penalized. Doing more work than agreed upon by the team and claiming extra contribution shall not be acceptable. Cooperation is essential!

Grading Sheet

