BSTMS team

Makerere university | kampala

Software Design Document

Black spot traffic monitoring system

Contents

[2 INTRODUCTION 1](#_Toc507098448)

[3 PURPOSE 1](#_Toc507098449)

[4 SCOPE 1](#_Toc507098450)

[5 OVERVIEW 1](#_Toc507098451)

[6 Reference Material 2](#_Toc507098452)

[7 DEFINITIONS AND ACRONYMS 2](#_Toc507098453)

[8 SYSTEM OVERVIEW 4](#_Toc507098454)

[9 SYSTEM ARCHITECTURE 4](#_Toc507098455)

[9.1 ARCHITECTURAL DESIGN 4](#_Toc507098456)

[9.2 DECOMPOSITION DESCRIPTION 5](#_Toc507098457)

[9.2.1 IP Camera 5](#_Toc507098458)

[9.2.2 BSTMS SERVER 5](#_Toc507098459)

[9.2.3 WEB SERVER 6](#_Toc507098460)

[9.2.4 MySQL DATABASE 6](#_Toc507098461)

[9.2.5 CLIENT WEB BROWSER 6](#_Toc507098462)

[9.3 DESIGN RATIONALE 7](#_Toc507098463)

[10 DATA DESIGN 8](#_Toc507098464)

[10.1 DATA DESCRIPTION 8](#_Toc507098465)

[10.2 DATA DICTIONARY 8](#_Toc507098466)

[10.2.1 Table1: camera 8](#_Toc507098467)

[10.2.2 Table 2: offense 8](#_Toc507098468)

[10.2.3 Table 3: penalty 8](#_Toc507098469)

[10.2.4 Table 4: road 9](#_Toc507098470)

[10.2.5 Table 5: Payments 9](#_Toc507098471)

[11 COMPONENT DESIGN 9](#_Toc507098472)

[12 HUMAN INTERFACE DESIGN 9](#_Toc507098473)

[12.1 OVERVIEW OF USER INTERFACE 9](#_Toc507098474)

[12.2 SCREEN IMAGES 10](#_Toc507098475)

[12.3 SCREEN OBJECTS AND ACTIONS 10](#_Toc507098476)

[13 REQUIREMENTS MATRIX 10](#_Toc507098477)

[14 APPENDICES 10](#_Toc507098478)

[14.1 Tool set and environment 10](#_Toc507098479)

# INTRODUCTION

This document contains the software design descriptions for Black Spot Traffic Monitoring System (BSTMS) project.

This document provides the details of how the BSTMS system should be built. The details are represented by using graphical notations such as viewpoints, use case models, sequence diagrams, class diagrams, object behavior models and other supporting design information.

# PURPOSE

This software design document describes the architecture and system design of the BSTMS. The approach taken to the design has been to create a high level component based design that ensures that all of the requirements are addressed by one or more components. The component based design approach supports both reuse and customization in the software implementation. Each component can be developed by the best suited team member such that they can use any desired detailed design approach available to them and define the interfaces that enable communications with other components.

# SCOPE

The System Design Document (SDD) describes how the functional and non-functional requirements recorded in the Requirements Document and the Concept document are transformed into more technical system design specifications from which the Black Spot Traffic Monitoring System will be built. The SDD documents both high-level system design and low-level detailed design specifications.

The SDD describes design goals and considerations, provides a high-level overview of the system architecture, and describes the data design associated with the system, as well as the human-machine interface and operational scenarios. The high-level system design is further decomposed into low-level detailed design specifications for each of the system’s components, including hardware, internal communications, software, system integrity controls, and external interfaces.

# OVERVIEW

This document describes the detailed design, data, architectural, user interface alike of the Black spot Traffic Monitoring System (BSTMS).

# Reference Material

|  |  |
| --- | --- |
| [1] | Sony, Sony, [Online]. Available: http://www.pro.sony.eu/pro/lang/en/eu/product/video-security-ip-cameras-minidomes/snc-em602rc/specifications/#specifications. [Accessed 12 12 2017]. |

# DEFINITIONS AND ACRONYMS

|  |  |
| --- | --- |
| **Term** | **Description** |
| **AMD** | 64 bit architecture series of computer processors |
| **Android** | An open-source operating system used for smartphones and tablet computers. |
| **ANPR** | a technology for automatically reading vehicle number plates |
| **API** | Application Programming Interface |
| **ASP.NET** | An open-source server-side web application framework designed for web development to produce dynamic web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services. |
| **Blackspot** | A location on a road that is under surveillance of the BSTMS |
| **Browser** | A computer program that can interface a user with the internet |
| **BS-CAM-XX** | camera at black spot number XX (BS-CAM-XX) |
| **BS-RD-XX** | Black Spot Number XX On road RD |
| **BSTMS** | Blackspot Traﬃc Monitoring System |
| **C#** | C sharp |
| **Computer Literate** | Some one that can use a computer system. |
| **CPU** | Central Processing Unit |
| **CSS** | Cascading Style Sheet |
| **CSS3** | cascading style sheet 3 |
| **FR-\*** | Requirement \*. where \* is a number greater than 0 |
| **FTP** | File Transfer Protocol |
| **GNU** | A Unix like operating system. That is free software |
| **GPRS** | General Packet Radio Service |
| **GUI** | Graphical User Interface |
| **HTML** | Hypertext Markup Language |
| **HTTP** | Hypertext Transfer Protocol |
| **HTTPS** | Secure Hypertext Transfer Protocol |
| **ICT** | Information and Communication Technology |
| **IDE** | Integrated Development Environment |
| **IEEE** | Institute of Electric and Electronic Engineering |
| **Internet** | A collection of computers, and software networked |
| **Java** | A general-purpose computer programming language designed to produce programs that will run on any computer system. |
| **JavaScript** | A programming language for client side scripting |
| **JSON** | JavaScript Object Notation |
| **JSP** | Java Server Pages |
| **Md5** | A widely used hash function that produces a 128-bit hash value |
| **MITM** | Man In The Middle |
| **MSSQL** | Microsoft Structured Query Language |
| **MySQL** | an open source relational database management system (RDBMS) based on Structured Query Language |
| **OCR** | Optical Character Recognition |
| **openCV** | A library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage and is now maintained by Itseez. |
| **PostgreSQL** | An open source relational database management system (DBMS) developed by a worldwide team of volunteers. |
| **Programming Language** | A set of rules and tools required to make computer programs |
| **Python** | A programming language for working with AI |
| **R Package** | Helper program written to extend functionality of R programming language. |
| **R Script** | Document that contains R code or instructions. Interpretable by the R engine. |
| **RTSP** | Real Time Streaming Protocol |
| **SDLC** | software Development Life Cycle |
| **SHA1** | Secure Hashing Algorithm 1. A cryptographic hash function |
| **SRS** | Software Requirements Specification |
| **Tensorflow** | an open source software library released in 2015 by Google to make it easier for developers to design, build, and train deep learning models |
| **URL** | Uniform Resource Locator |
| **USB** | Universal Serial Bus. |
| **VCS** | Version Control System |
| **Windows** | operating system by Microsoft corporation |

# SYSTEM OVERVIEW

The BSTMS shall rely on openCV computer vision library to analyze video streams of moving cars. The video shall be captured using an IP Camera. The cameras shall be to be near the target black spots. The cameras shall send/stream the recorded video to a BSTMS server that shall analyze the video stream for traffic offenses vehicles. The server shall record the data into a database for the streaming applications from the oﬃcers to access it. Figure 1 shows the way the system shall operate.

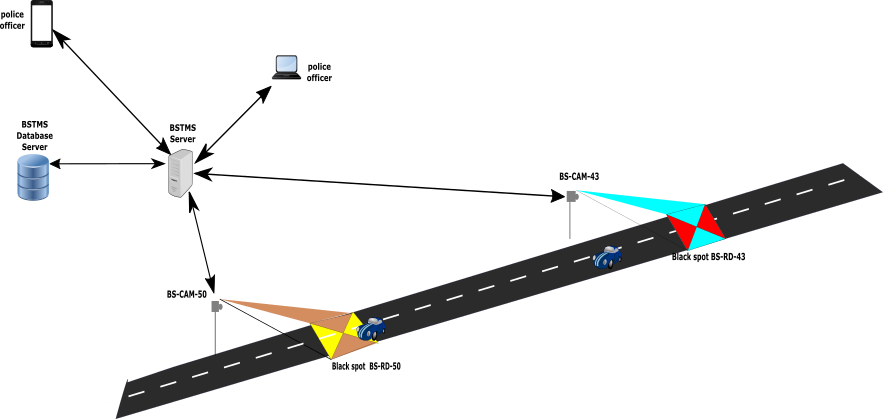


Figure : Overview of Black Spot Traffic Monitoring System

# SYSTEM ARCHITECTURE

## ARCHITECTURAL DESIGN

The BSTMS system is broken up into three major components:

A client-side Web application and a server-side Python Console application and MySQL database.

The client-side application is also separated into two parts: the functional component (written in Java), and the graphical component (written in HTML). The functional component forms the core of the BSTMS web application. It receives user input and constructs groups, bills, and transactions. It performs all of the calculations required to resolve debts.

The graphical component, as the name implies, is simply the graphical user interface. It provides all of the buttons, text boxes, and other on-screen elements which allow the user to access all of the features provided by the application.

The server-side component of BSTMS is comprised of a Python interface, which manages incoming video streams from cameras, analyses each stream for traffic offenses, records a photo evidence of the offense and communicates with the database.

The MySQL database provides centralized storage for synchronized data.



## DECOMPOSITION DESCRIPTION

This part includes a hierarchical structure of entities. Each entity is determined and briefly mentioned in this section. For each entity a detailed description will be provided in the sixth section. It can be seen entity diagram of overall system in the [component design section](#_COMPONENT_DESIGN)

### IP Camera

This component shall capture videos and stream them to the BSTMS server. The IP camera shall be added by the administrator using the administrator page on the Web Client.

### BSTMS SERVER

#### IP Camera Streaming

This component shall capture and manage all video streams simultaneously and pass each video stream from each camera to the frame processing component.

#### Frame Processing

This component shall process the video stream to extract individual frames for the car recognition component.

#### Car Recognition

This component shall recognize cars in the video stream

#### Number plate Detection

This component shall detect a number plate from the image given

#### Traffic Offense detection

#### Offense Penalty

This component shall give penalty to the car license plate the has committed the ofense

### WEB SERVER

#### Video streaming

The web server shall allow streaming of video of a selected camera via a web socket.

#### Client update

This component shall update the user’s browser with new content.

#### Authentication

This component shall handle authentication of the administrator to access critical components of the system.

#### Session Management

This component shall handle sessions of the administrator

### MySQL DATABASE

This component shall persist records of the traffic monitoring as described in the data design section of this document.

### CLIENT WEB BROWSER

The client web browser shall be any common web browser capable of supporting HTML5 and CSS3. The browser shall be the main mode of interaction with the BSTMS. The component shall send HTTP requests to the Web Server that shall reply with HTTP response of requested information.

#### Video Streaming

This component shall allow the users to view live video of the camera doing the traffic analysis.

#### Add Camera

This component shall allow the administrator to add a camera to the system.

#### Edit Camera

This component shall allow the administrator to edit a camera’s features.

#### Delete Camera

This component shall allow the administrator to delete a camera

#### View Analysis

#### View Help

This component shall allow all users to access help pages of the system.

#### Login

This component shall work with the authentication component to allow the administrator to login to the BSTMS server.

#### Print Report

This component shall allow the administrator user to print the traffic analysis report

#### Pay Penalty

This component of shall be requested for by the client web browser. The web server then responds with a penalty payment form. The user enters a few details and submits the form.

## DESIGN RATIONALE

In this project, design choices are made according to performance concerns and integrality of the system. System has to be designed in a way that future models and features can be added and current models can be changed and updated independently.

Stakeholders may have and request further requirements, therefore system parts have to be modular. Developers of the system has to document development process and use comments in their code frequently, so that in the future other developers may understand code and the structure of the system

In this project, Unified Modeling Language (UML) is selected as a part of design viewpoint and it will be used for clarifying design viewpoints. Developers of this system shall use UML notation in naming and writing of source code.

# DATA DESIGN

## DATA DESCRIPTION

MySQL database and JDBC to communicate with the database that is installed locally on the server.

## DATA DICTIONARY

The data dictionary for the database of the BSTMS

### Table1: camera

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| ID (Primary) | int(11) | No |  | Auto increment value |
| Name | varchar(40) | No |  | Name of the camera |
| roadID | int(11) | No |  | Id of the road |
| url | Text | No |  | URL which contains camera stream |
|  |  |  |  |  |

#### Table 1: Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | ID | 0 | A | No |  |
| roadName | BTREE | No | No | roadID | 0 | A | No |  |

### Table 2: offense

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| Type | int(1) | No |  | Type of offense 1,2 or 3 |
| date\_committed | datetime | No | CURRENT\_TIMESTAMP | Date and time of commission |
| number\_plate | varchar(20) | No |  | Number plate of offender |
| camera\_ID | int(11) | No |  | ID of camera that captured incident |
| penaltyID | int(11) | No |  | ID of penalty attached to offense |
| ID (Primary) | int(11) | No |  | Auto increment value |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | ID | 0 | A | No |  |
| camera\_ID | BTREE | No | No | camera\_ID | 0 | A | No |  |
| penaltyID | BTREE | No | No | penaltyID | 0 | A | No |  |

#### Table 2: Indexes

### Table 3: penalty

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| ID (Primary) | int(11) | No |  | ID of the penalty |
| Description | text | No |  | Details on the penalty |
| Amount | double | No |  | Fee charged for the penalty |
| Name | varchar(20) | No |  | Name of the penalty |

#### Table 3: Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | ID | 0 | A | No |  |

### Table 4: road

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| Name | varchar(65) | No |  | Name of the road |
| ID (Primary) | int(11) | No |  | ID of the road, auto increment |
| from\_ | varchar(65) | No |  | Where the road begins |
| to\_ | varchar(65) | No |  | Where road ends |

#### Table 4: Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | ID | 0 | A | No |  |

### Table 5: Payments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| Amount | int(20) | No |  | Type of offense 1,2 or 3 |
| date\_paid | datetime | No | CURRENT\_TIMESTAMP | Date and time of commission |
| number\_plate | varchar(20) | No |  | Number plate of offender |
| offense\_ID | int(11) | No |  | ID of camera that captured incident |
| penaltyID | int(11) | No |  | ID of penalty attached to offense |
| ID (Primary) | int(11) | No |  | Auto increment value |

#### Table 5: Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | ID | 0 | A | No |  |
| offense\_ID | BTREE | Yes | No | offense\_ID | 0 | A | No |  |
| penaltyID | BTREE | Yes | No | penaltyID | 0 | A | No |  |

# COMPONENT DESIGN

*In this section, we take a closer look at what each component does in a more systematic way. If you gave a functional description in section 3.2, provide a summary of your algorithm for each function listed in 3.2 in procedural description language (PDL) or pseudocode. If you gave an OO description, summarize each object member function for all the objects listed in 3.2 in PDL or pseudocode. Describe any local data when necessary.*

# HUMAN INTERFACE DESIGN

## OVERVIEW OF USER INTERFACE

The used GUI components are menus, buttons, textboxes, down drop lists, links, and tables. The main means of user interface shall be through a web-browser. The BSTMS system shall be a Web application. Only the administrator shall login to the system. All other users only access the web application.

The administrator shall have interfaces to login, add cameras, and monitor cameras, view analysis of camera.

All non-administrative users shall have interfaces to view offenses, view/stream a camera.

## SCREEN IMAGES

### Dash board

This shall be the landing page for all users and enable quick access the system

### Add Camera

This shall allow the administrator to add a camera t the system.

### View Help

This shall contain help information on how to use the system.

### Stream Video

This shall allow the users to stream a video of the selected camera

### Pay Penalty

This shall allow traffic offenders to pay penalties given by the system.

### View Report

This shall allow all users to view the traffic offenses report.

### Offense Penalties

This shall allow the administrator to add/view/edit traffic offense penalties.

### Penalty Payments

This shall allow the administrator to view information of all penalties paid.

### Traffic Offenses

This shall allow all users to view traffic offenses detected at each camera.

Display screenshots showing the interface from the user’s perspective. These can be hand­ drawn or you can use an automated drawing tool. Just make them as accurate as possible. (Graph paper works well.)

## SCREEN OBJECTS AND ACTIONS

*A discussion of screen objects and actions associated with those objects.*

# REQUIREMENTS MATRIX

*Provide a cross­reference that traces components and data structures to the requirements in your SRS document.*

*Use a tabular format to show which system components satisfy each of the functional requirements from the SRS. Refer to the functional requirements by the numbers/codes that you gave them in the SRS.*

# APPENDICES

## Tool set and environment

|  |  |
| --- | --- |
| Tool | Purpose |
| Microsoft Visio 2010 | UML diagrams: block diagram, DT, Sequence diagrams, Activity diagrams, Class diagrams |
| Microsoft Office Word 2013 | Documentation |
| MySQL | Relational Database Management |
| Microsoft Office Power Point 2013 | Presentation |
| Tomcat 8 | HTTP Servlet Container |
| Pycharm-community Edition | Python Programming Environment |
| Netbeans 8.2 | JavaWeb (Web Application) Programming Environment |
| Bootstrap 3 | CSS Framework |
| JQuery | JavaScript Manipulation Library |
| OpenCV 3 | Computer Vision Library To Analyse Video streams in the System |
| Open ALPR | License/Number Plate Recognition |
| JavaFX Scene Builder | GUI Construction For Administration Of System |
| Xubuntu 16.04.4 | Operation System in Programming |
| Notepad++ | Quick Code Preview in Windows |
| Mousepad | Code preview/editor in xubuntu |
| CMake | Compiling make files for openCV 3 on Ubuntu |
| GNU Make | The make utility automatically determines which pieces of a large program need to be recompiled, and issues commands to recompile them. |
| Python 3 | The main Programming Language for the BSTMS |
| OpenCV-python | Prebuilt python binaries and library for openCV |