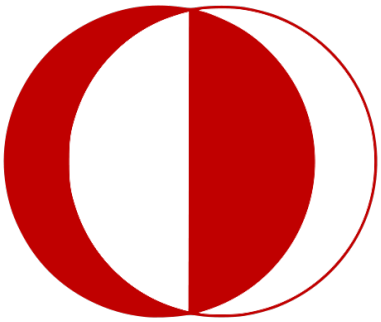
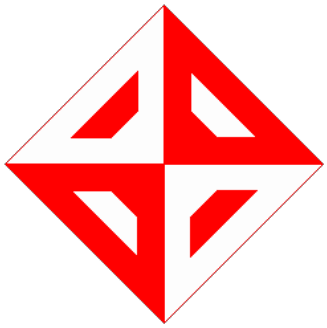
Middle East Technical University

Department of Computer Engineering





CONVEYOR

Software Design Description Document

V1.1

Arctic Donkeys

Zeynep Miray Mazlumoğlu - 1819481

Arda Aslan - 1881010

Göksucan Akın - 1818905

Onur Yılmaz - 1819671

01.03.2015

Change History

Date	Revision	Comment
04.01.2015	1.0	Created
01.03.2015	1.1	3.2.2. Influences on Software Life Cycle Product is updated. 4.3. Design Stakeholders and Their Concerns is updated. 4.8 Design Rationale is updated. 5.2.2. Design Elements is updated. 5.4. Logical Viewpoint is updated. 5.6.2. Design Elements is updated. 5.8.2. Design Elements is updated. 5.10. Information Viewpoint is created.

Table of Contents

1. Overview

1.1. Scope

This Software Design Description document illustrates the comprehensive structure of components of the Conveyor. Design and development process involves design elements and design viewpoints. These design elements are based on IEEE Std 1016 Software Design Descriptions template. In order to increase the understandability of these design aspects, appropriate UML diagrams will be provided in a detailed way. Document is prepared to provide essential design and implementation features to the stakeholders who are interested in the design process of Conveyor system.

1.2. Purpose

The software design document presents the system modelling and architectural design for Conveyor System. The purpose is to describe the design elements, relationships of entities and system components and to create a general description. This eases implementation level behaviour of objects and clarifies the coding phase. Software Requirement Specification document of Conveyor system is highly related with this design document. Interactions between components and individual behaviors of components are stated in SRS document. While preparing the design report, many sections will refer to SRS diagram so, obedience of SDD to SRS is really essential. In addition to SRS properties, SDD also gives an implementation level integrity and focus on mostly design issues. . While implementing the Conveyor system, this SDD document is expected to be highly used by stakeholders.

1.3. Intended Audience

As stated in section “1.2. Purpose” of Conveyor system SRS Document, disaster relief organizations, emergency managers, civil administrators and people who live in seismic areas are intended audience for this project. This document is prepared for knowledgeable project

managers, developers and testers by explaining data, procedural design and architecture(design viewpoints) in detail. The end users can also be guided by system boundaries and services description in this document.

1.4. References

i. ArcGIS Desktop. (2010, July 23). Retrieved November 28, 2014, from http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//006m00000069000000.htm

ii. "Conveyor" Software Specification Requirements (SRS) Document. (2014).

iii. Mann, K. (2012, January 1). ArcUser. Retrieved November 28, 2014, from http://www.esri.com/news/arcuser/1012/a-workflow-for-creating-and-sharing-maps.html

iv. StarUML 5.0. User Guide. (2005). Retrieved from http://staruml.sourceforge.net/docs/user-guide

v. IEEE Standard Documents:

1. IEEE. (2009). IEEE Standard for Information Technology-Systems Design-Software Design Descriptions.IEEE Computer Society.

vi. UML 2 Composite Structure Diagram. (n.d.). Retrieved January 3, 2015, from http://www.sparxsystems.com.au/resources/uml2_tutorial/uml2_compositediagram.html

2. Definitions

Term	Description
SDD	Software Design Description
SRS	Software Requirement Specification
UML	Unified Modelling Language
Conveyor	Name of the system
GIS	Geographic Information System
ArcGIS	GIS Software to be used
MVC	Software architectural pattern
JAVA	Programming language to be used

Eclipse	Multi-language integrated development environment
DBMS	A database management system is a collection of programs that enables developer store, modify and extract information from a database
Hadoop	Open source software framework for storage and large-scaling processing of data sets
MySQL	Open source relational database management system
Admin	Person who makes create, delete and update operations to database of the system
Client	Person who sends query by using the system
Scrum	Iterative and incremental Agile Software development framework for managing product development
AKUT	Search and Rescue Association
ER	Entity Relationship Diagram
CRUD	Create, Read, Update and Delete

3. Conceptual Model For Software Design Descriptions

This section includes basic project terminology and concepts of SDD. In addition to these, the context in which SDD is created and used is also included. The stakeholders who use them and how they are used are introduced. By this way, a detailed concept is provided to enhance understandability of software life cycle, interactions of components and frameworks that the system uses.

3.1. Software Design In Context

The Conveyor System is created in MVC pattern. Model View Controller pattern is widely used in products because of the fact that changes in one representation are reflected to all views. This

software product is designed by using object oriented approach. Java programming language is used while implementing this fashion. Eclipse (Luna) is sought as an IDE.

Moreover, for more detailed graphical result, this project uses ArcGIS for visualization. Any device that operates with Java and ArcGIS can use and run the provided software. The Conveyor will be input dependent since the information of an arbitrary earthquake and the name of the location to be displayed has to be entered. As an important issue, optimizing the performance of searches and generating output is needed. The determining factor that shows the performance of our project will be the size of the data for a given location. The data of a location is consisted of building properties and the earthquake impact on these buildings. MySQL is used to store the relevant data about the project.

3.2. Software Design Descriptions Within the Life Cycle

3.2.1. Influences on SDD Preparation

The main influence on software design process is the Conveyor System SRS Document.

The project is designed in the light of functional and nonfunctional requirements, the product perspective and the interface requirements in the SRS. Since agile method is endorsed as a development method, stakeholders can specify extra requirements in the development process. SDD document is convenient for these extra requirements and can be modified at further versions.

3.2.2. Influences on Software Life Cycle Products

The requirement specifications of stakeholders can change throughout the development. (Either a stakeholder adds a new specification or a non-implementable requirement was specified earlier). Currently there are not any major changes on SRS caused by SDD.

Test plans for the Conveyor will be generated and SDD will influence these test plans. So, test documents should consider SDD as their base reference.

3.2.3. Design Verification and Design Role In Validation

Software design description is the primary decision mechanism for the verification and validation whether the system satisfies the requirements in the Conveyor System SRS Document and stakeholders' design concerns. Furthermore, the SDD is determinative whether the system conforms the exposed architecture. Since the SDD affects the test cases, test plans and test cases are prepared after the code development.

4. Design Description Information Concept

4.1. Introduction

Software Design Description of Conveyor System establishes system structure and how it will be implemented. While doing this, some prior design viewpoints ,which includes context, composition, logical, dependency, information, patterns use, interface, structure, interaction, state dynamics, algorithm and resource viewpoints, and their design concerns will be explained in detail by the help of UML diagrams. Stakeholders and their roles will also be discussed extensively. In order to give an idea about system design, prototype of the system supported by interface and GIS sample output illustration will be visualised. In addition to these, design rationale, design overlays and design views will be expressed in later sections.

4.2. SDD Identification

The system design illustrated in this document presents the main document of the Conveyor. This design will be used through the project ;in other words, system model defined in this document will be used in each iteration of implementation and development periods. All system and implementation changes will be reflected as an iteration on this document. After initial iterations, the prototype will be demonstrated on January 19, 2015. The software product will be released towards the end of May, 2015. The tentative day for the date of issue is May 28, 2015. Arctic Donkeys team will be responsible for issuing the Conveyor System with the issuing organisation SAP mentor Onur Deniz. All rights of the end product are reserved. Both Arctic Donkeys team members and SAP mentor are free to alter and release the product due to the exclusive rights subject. However, because of the copyright of the system data collected from

Turkish Statistical Institute, Metu Disaster Management Implementation and Research Center cannot be distributed or shared.

Scope, references, context and summary can be examined in section 1.Overview. Abbreviations, definitions can be referred in section 2.Definitions part. Glossary part can be found at the end of the document while change of history can be examined in page 1. Other than these, conceptual model for software design descriptions can be found in section 3. Details of the project including several viewpoints are extensively examined in section 5.

UML design language will be used for visualising the viewpoints.

4.3. Design Stakeholders and Their Concerns

Stakeholders consist of Arctic Donkeys software and system development team members, SAP mentor Onur Deniz and end users mainly disaster relief organizations, emergency managers, civil administrators and people who live in seismic areas. Testing will also be performed by Arctic Donkeys team. Conveyor is mainly based on recommend an applicable emergency plan. The main concerns of the end users are creating most accurate model ,that covers predicting collapse risk of the buildings and distributing AKUT sources considering risk priority, using existing data and visualising model on GIS interface in understandable way. Arctic Donkeys development team is addressed to implement and test these needs. SAP mentor is responsible for giving idea about implementation and used technologies. The end product will cover all main concerns of end users. Details of the implementation will be discussed in later sections. For further details about the problem, please refer to SRS document of the Conveyor.

4.4. Design Views

The Conveyor will be implemented as a user input and data dependent desktop application with a simple java interface. Results will be shown in advanced Graphical Information System interface.

The SDD of the system includes several viewpoints. These viewpoints contain contextual, compositional, logical, informational and structural views and also dependency, state dynamics, interfaces, used patterns, algorithms and resources. When distinct viewpoints are provided to users and stakeholders, the aim and structure of design are correctly observed. By these different viewpoints, stakeholder observes different characteristics of the design in detailed way. For a complete design these viewpoints are crucial. If the interactive testing of different viewpoints does not create controversy, the system becomes consistent.

4.5. Design Viewpoints

Conveyor System SDD contains context, composition, logical, information, pattern, interface, structure, interaction, state dynamics and algorithm with resource viewpoints. In context view, the interaction of system services with users will be examined by the help of use case diagrams and their explanations. Which user can perform which actions will be the key concept of this viewpoint. Secondly in compositional view, functional decomposition as well as run time (physical) decomposition will be explained by using component and deployment diagrams. System is divided into subcomponents and their relations will be shown. Thirdly, logical and structural view contains static structure of system with reuse of types and implementations. Class diagrams will help in that case since it contains classes, attributes, functionalities and relationships between themselves. Information view gives a clue about persistent information. When subject comes to persistency, data first comes to mind. Entity-relation diagrams visualise the data and its attributes along the interaction with other data. Reuse of pattern with suitable framework construction creates another view. Component diagrams and template interface designs give stakeholders an idea about service accesses and system role. Interaction view shows the communication and flow of operations via sequence diagrams. Dynamic state transformation will be explained with state diagrams. Algorithm issue and resource utilisation contains the details of the algorithm and the resource usages. All viewpoints will be examined in section 5 in detail.

4.6. Design Elements

A design element can be described as any item occurring in a design view. These items have a name, type and contents. In the following sections, these elements, which can be design entity, relationship, attribute or constraint, and their viewpoints will be explained deeply.

4.6.1. Design Entities

A design entity represents the key elements of a software design. If necessary to exemplify the design entities, the examples can be listed as systems, subsystems, abstract collaboration patterns, patterns, generic templates, classes, components, libraries, frameworks and programs. This list can be extended easily.

4.6.2. Design Attributes

The attribute of that entity includes the name, type, purpose and author of a design entity.

4.6.2.1. Name Attribute

Since the name of attribute is used to determine the purpose of the element, each design element shall have a unique name to avoid the misunderstanding of the purpose.

4.6.2.2. Type Attribute

Each design entity shall be classified. While understanding behaviour of the entity, the type of entity plays an important role.

4.6.2.3. Purpose Attribute

Importance and purpose of the entity shall be explained in a few words to understand or determine the aim of that design element.

4.6.2.4. Author Attribute

If it is needed to change the name, type or purpose of the design element, communicating with the author of the element has a vital importance. So, the author of the element shall be identified clearly.

4.6.3. Design Relationships

While defining the relationships between design elements, the composition of them can be used. This relationship titles as an association or correspondence among two or more design entities. The relations between design entities shall be defined with name, type, purpose and author like a single design element.

4.6.4. Design Constraints

Like design relationships, design constraints which are the last element of the design view also have name, type, purpose and author. The design constraints specify the restrictions applied by the source element to the target element that can be design attribute, relationship or entity.

4.7. Design Overlays

The interface viewpoint contains the user interface of the Conveyor. This user interface should not be confused with the ArcGIS user interface. The user interface of this project refers the interface of the Conveyor. Its main purpose is receiving inputs to simulate a fictitious earthquake and displaying the estimated scenario.

4.8. Design Rationale

Conveyor System is designed for recommending an emergency plan for aimed profile -please refer to section 4.3- and it is data based desktop project. Collecting real data of the buildings, earthquakes and target area ground information presents the challenging and most important part of the project since the Conveyor is based on data processing and giving approximate model created using that data. Modeled emergency plan covers the risk map of the pilot area according to given inputs and distribution of the resources considering risk map. Since our project draws

triangle between data, modeling algorithm and visualisation of the combination of them, MVC architectural pattern is most suitable pattern for Conveyor which offers a clear connection between updating GIS related to changing data-model combinations. In other words, MVC enables developers to reflect immediate changes of one user to all users. Data composes model, ArcGIS composes view and Modeling algorithm composes controller part of the architecture.

MySQL server is used to store data and other database operations. Connection between database and GIS interface is crucial in terms of visualising the data in fast manner. Since MySQL is compatible with ArcGIS environment, choosing MySQL as an database server is reasonable. The only bottleneck of the MySQL is storing data in row-based style. This may reduce efficient data operations. However, team is working on data storing to make faster operations. Some database management tools offers column based operations; however, these are not free. SAP HANA platform may be suitable for the project but licensed version is needed. Optimizing data operations on free and common used server is the most suitable way for the developers for now.

This product will be implemented by using Java Programming Language. Since it enables programmers object oriented coding and many high level technologies supports this language, it will be appropriate for the product. Regression testing, optimizing algorithms can be easily implemented on Java and it serves better performance comparing with other programming languages. Moreover, interface creation is user friendly in Java and it offers better visualisation options. Other than Java, newest and effective approaches will be used because they serve better utilisation and performance.

It is said that performance is important as much as correctness. To improve the performance, team is working on database operations. Other than this, code revising will be done to maximise efficiency. Using complicated structure will be avoided because of the same reason.

While creating synthetic part of the data, district borders are needed. To detect approximate border, matlab is used to get best precision and most suitable equations.

This is out of agreement for now but for big data issue, Hadoop may be integrated to the DBMS. Thus, handling massive data becomes easier.

4.9. Design Languages

UML and ER diagrams are selected as a part of design viewpoint specification. UML are used for the classes and function definitions and ER diagrams are used to represent the entity relationships.

5. Design Viewpoints

5.1. Introduction

Design viewpoints compose the principles of a system including the architectural models, languages and notations. While determining the design descriptions and limitations of the Conveyor, these principles are used. Context, composition, logical, dependency and information viewpoints are the primary viewpoints will be mentioned in this part of SDD. However, patterns use, structure, state dynamics, algorithm and resource viewpoint will be discussed in this part of SDD, as well.

5.2. Context Viewpoint

In contextual design, system interactions between user and stakeholders are visualised by using use cases through UML. The relationship between internal and external components of Conveyor System is displayed by context diagram. Definitions of roles and interactions of system will be discussed in following subsections.

5.2.1. Design Concerns

Conveyor system has two different user types and abilities of each are discussed in related use case diagrams. One of them is client and s/he has limited abilities. Client can send query through the provided interface of the Conveyor system. This query consists of latitude, longitude, magnitude and focal depth of the earthquake to be simulated and a district value for visualization process. This request made to user interface is reflected to Conveyor system's database. After applying appropriate modelling algorithms and sequence of processes, response of the system is reached among the user interface of ArcGIS to the client. Second user type is admin. Admin has more authoritative abilities which are directly related with database. An admin can manipulate

the database by performing CRUD operations. There are three more components of the system environment. Two of these are user interfaces. One is the interface of Conveyor system and the other one is the interface provided by ArcGIS. As it is stated, Conveyor interface is for demanding input from the user and ArcGIS interface is for visualizing the output. Last but not the least vital component is the database as the project is based heavily on data. Figure 1 illustrates the configuration of the Conveyor system among user interfaces and database within users.

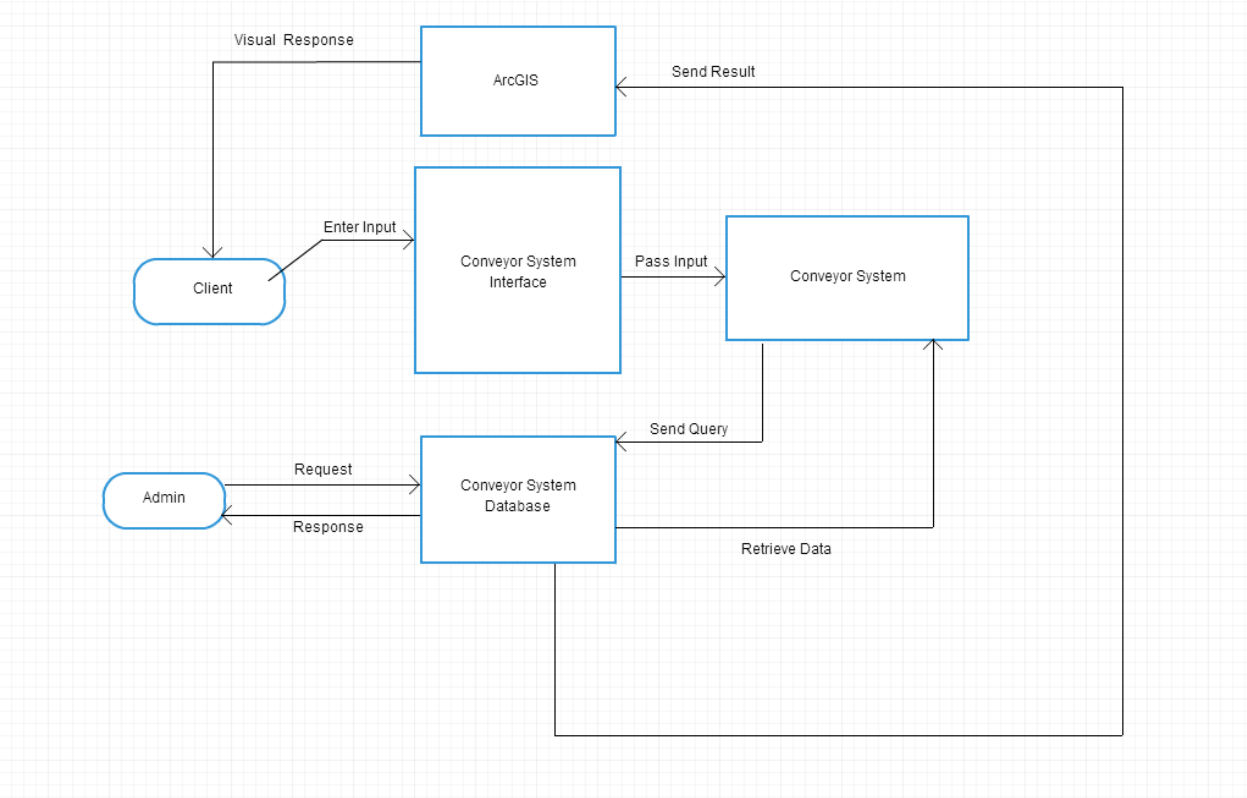


Figure 1 - System Context Diagram

5.2.2. Design Elements

One of the major design element is the group of stakeholders. The Conveyor system is a software project conducted in association with Onur Deniz from SAP company. The stakeholders are Conveyor system development team members.

Other design elements are design entities which are actors, developers, subsystems and database. All entities are investigated throughout the section

Name: Client

Type: Actor

Purpose (Function attribute): Client entity is one of the most important entities in this system. System requires user-entered data. Queries are composed according to these data and sent to database. Also the main purpose of the system is to provide a recommendation to people who want to be informed about the effect of a specific earthquake to a region. So it can be said that the main audience of the project is clients and the system can be considered as user oriented system.

Relationships: Clients can interact with two interfaces separately. First is the interface of Conveyor system. Client need to enter some data about the earthquake to be simulated into the appropriate boxes of the user interface. After this operation, information is sent to database through Conveyor system. Second is the ArcGIS interface. When a user requests a modelled output, Conveyor system informs the ArcGIS with a modelled output. Then, ArcGIS displays that output to the client. Other than this demonstration process, client can choose a point from the map if s/he wants a path from the nearest rescue agent to be drawn to the chosen point.

Constraints: Entered inputs from user should be in provided units and there shouldn't be any empty fields. For Latitude text box, valid entry should be between 40.28-41.33. For Longitude text box, valid entry should be between 28.01-29.55. For Magnitude text box, valid entry should be between 4-10. For Focal Depth text box, valid entry should be between 0-200.

****

Name: Admin

Type: Actor

Purpose (Function attribute): Main purpose of an admin is to manipulate the database in order to provide factual data to the system.

Relationships: Admins can only interact with the database. Any admin can perform create, read , update and delete operations on the database.

Constraints: Any manipulation to the database shouldn't deform the database tables.

****

Name: Developer

Type: Actor

Purpose (Function attribute): Developers take role in nearly all design viewpoints. They are not directly related with the usage of the system. They implement all aspects of the system. Without developers the system cannot be completed. In context viewpoint, developers interact with all design entities.

Relationships: Developers interact with all design entities. Important interactions will be declared in further versions.

Constraints: There are some main constraints on the system and they concern the developer. Developer shall design the project compatible with Windows operating system. Another constraint is that providing a usable and understandable user input interface. For reusability concern, developers should write comprehensible code. They shall use Eclipse Luna as an IDE.

****

Name: ArcGIS

Type: Subsystem

Purpose (Function attribute): ArcGIS system presents the modelled data to clients. So “data visualization” part is committed by ArcGIS subsystem.

Relationships: ArcGIS subsystem interacts with database and clients. Client interaction is explained in relationships section of “client” entity. ArcGIS’ goal is to store display the modelled data. The only interaction with database is the fetching operation of data to be displayed.

Constraints: There are no constraints for ArcGIS

****

Name: Conveyor

Type: System

Purpose (Function attribute):Conveyor system holds all of the algorithm development and internal processes. Graph traversal, risk calculation algorithms and user interface are implemented here.

Relationships: Conveyor system interacts with database and clients(through user interface). Client interaction is explained in relationships section of “client” entity. Conveyor’s goal is to create a model and apply graph traversal algorithm by the help of the query from the client. It composes that query and data from the database to create a model.

Constraints: Conveyor system shall be compatible with Windows OS. Output shall be correct and reliable. The database will be MySQL. ArcGIS shall be used for visualizing the model. Lastly the response time of the system must be at most 10 minute.

****

Name: MySQL

Type: Database

Purpose (Function attribute): Since the main source of the project is data, a database is required. MySQL is used as a database. It supports SQL. The entered data is stored in MySQL and desired data to be processed is fetched from MySQL too.

Relationships: MySQL interacts with ArcGIS and Admin. All of these interactions are stated in previous sections.

Constraints: There are two vital constraints for database. First is that, variables with string types will be at most 255 characters long. Lastly, coordinate will be represented as long double. Types for other variables are defined in class diagram and database tables shall be prepared accordingly.

****

The visualisation of users and their functionality relations are described in Figure 2. The detailed use case descriptions are provided in Appendix which are taken from Conveyor System SRS Document.

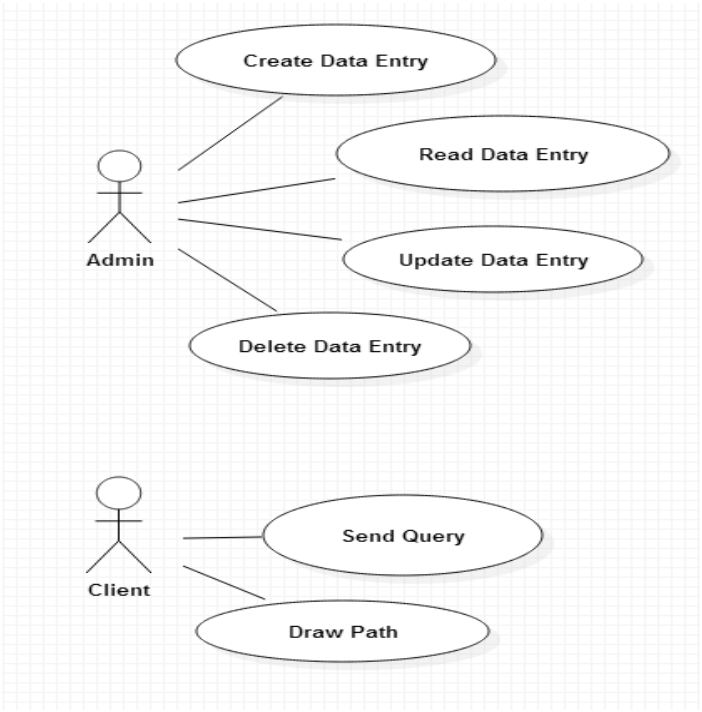


Figure 2 - Use Case of Admin and Client

5.2.3. Example Languages

UML design language is used to describe use case and context diagram. Related schemas and use cases are used in 5.2.1 and 5.2.2 sections.

5.3. Composition Viewpoint

This section provides information Conveyor system components and their connections with each other. The correlation between the whole system and the Conveyor system as a desktop application is also shown inside the overall system environment.

5.3.1. Design Concerns

The aim of this viewpoint is providing information to stakeholders and programmers for planning and controlling the system. This kind of subsystem level illustration can be used for assembling components, cost estimation and schedules in terms of development effort.

System components such as libraries, packages, files and their interconnections are illustrated in UML Component diagram. Conveyor System is integrated with the visual system which is ArcGIS. ArcGIS will be the one drawing the map according to the data given to it to print out a general hotspot map according to the risk percentages of chosen area. Clients can send queries through the Conveyor system via their PC which is compatible with Windows operating system to see the possible danger zones in a possible earthquake case in order to pre-determine the organization of help services for this arbitrary disaster. These queries will be processed in the Conveyor system with data fetched from database.

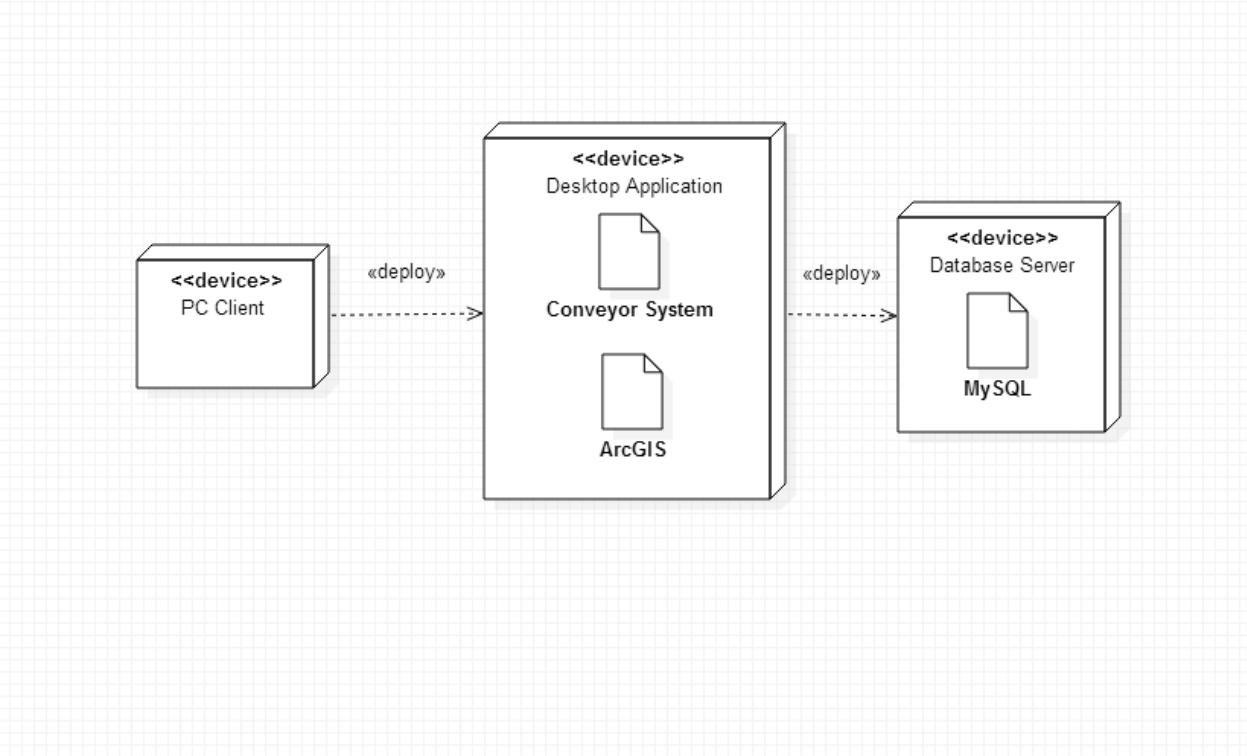


Figure 3 - Deployment Diagram of Conveyor

5.3.2. Design Elements

Design elements are design entities which are algorithms, desktop application, database and visualization layer and Conveyor system user interface. All entities are investigated throughout in section 5.6.2 and 5.5.2. Related component diagram is provided in section 5.6, please refer to this section.

5.3.2.1. Function Attribute

Algorithms

Algorithms is used to provide an algorithm that is to be used while calculating risk of collapsing and drawing path to destination.

Desktop Application

Desktop application is used to provide communication between algorithms and database layer in terms of data generation and modelling an collapsing rate of an building in an earthquake situation.

Database Layer

Database layer is used to store information about building, former earthquake, rescue agent and fault line.

Visualization Layer

Visualization layer is used to visualize ArcGIS hot spot map which is drawn with respect to calculated risks and user interfaces of Conveyor.

Conveyor System User Interface

Conveyor system user interface controls the changes in information and displays them coming from ArcGIS.

5.3.2.2. Subordinates Attribute

Algorithms

Algorithms takes information of building, faultline and rescue agent from database and a model from desktop application to calculate collapsing risk of building. On the other hand, while drawing path to destination, algorithms takes place of departure point from user interface and destination point from ArcGIS and provides a recommended path.

Desktop Application

Desktop application handles two operations, which are data generation and creating model. In data generation, admin generates data about the building, former earthquake, rescue agent and district, and then, these generated datas are written to database. In creating model, admin creates own collapsing building in earthquake situation model to be able to use in algorithm part.

Database Layer

Desktop Application connects to MySQL Database and store provided information from the user interface in database.

Visualization Layer

Visualization layer consists of Conveyor System User Interface component and ArcGIS subcomponent. ArcGIS shows result of selected the hot spot map of districts to users. Related information about Conveyor System User Interface will be provided in next section.

Conveyor System User Interface

Conveyor system user interface takes district name and latitude, longitude, magnitude and focal depth about earthquake from user and sends query to database.

5.3.3. Example Languages

UML modelling language is used while constructing deployment and component diagrams explained in 5.3.1 and 5.3.2 sections. These diagrams are relevant for dependency and structure viewpoints. These diagrams are relevant for dependency and structure viewpoints.

5.4. Logical Viewpoint

This section provides implementation clues of the system by defining class diagrams. By this diagram, static structural relationships and interface design are defined. Roles of the parts of the classes are explained in following sections.

5.4.1. Design Concerns

Conveyor is highly data dependable system. The size and the organization of the data is important while constructing the model since it is determined by statistical processing of the data. Creating realistic attributes is crucial in terms of manage the modeling algorithm. Whole data used in Conveyor system is gathered by own effort. Relationships , attributes are determined by Arctic Donkeys' team members. Details of the data attributes will be given in next section. In addition, data organization and passing the objects between different classes and their functionalities will be examined in next section.

Logical viewpoint is used for clarifying argument function relationships. Hence, it gives development and reuse information about class implementations and its attributes. The whole of class implementation is aimed to return correct results to the Conveyor System user specific to given input functionality. The class structure of system determines the implementation and development parts of the project. In implementation part of process, the changes in class diagrams may occur. In case of change in classes, they will be reflected in updated documents.

5.4.2. Design Elements

Class Diagram Explanations

Building:

Name	Type/Parameter	Explanation
location	Pair: float, float	Each building has unique location in terms of latitude and longitude
age	int	Age of building
numberOfLivingPerson	int	How many people living in building
ground	String	Ground type of building
depthToMainland	float	Depth of building to mainland
isDamaged	boolean	Whether building is damaged or not
risk	float	Collapse risk of building
districtName	String	District name of building

Table 1 - Explanation of Building Class

Rescue Agent:

Name	Type/Parameter	Explanation
name	String	Each rescue agent has unique name for identification
capacity	float	Capacity of building
location	Pair: float, float	Location of rescue agent in terms of latitude and longitude

Table 2 - Explanation of Rescue Agent Class

District:

Name	Type/Parameter	Explanation
name	String	Each district has unique name for identification
districtMap	WayMap	Map of the district

Table 3 - Explanation of District Class

WayMap:

Name	Type/Parameter	Explanation
districts	ArrayList<String>	This ArrayList contains the all district names in İstanbul
beginEndPoints	ArrayList<Pair: float, float>	This ArrayList contains the starting and the endpoint coordinates of the district's streets.

Table 4 - Explanation of WayMap Class

Faultline:

Name	Type/Parameter	Explanation
type	String	Type of fault line with respect to rate
length	float	Length of the faultline
isActivated	boolean	Whether this fault line is activated or not

Table 5 - Explanation of Faultline Class

Earthquake:

Name	Type/Parameter	Explanation
date	Date: String	Each earthquake has unique date for identification

districtName	String	Name of the district the earthquake happened
location	Pair: float, float	Location of the earthquake
focalDepth	float	Depth of the earthquake
magnitude	float	Magnitude of the earthquake
faultLine	Faultline: String	Name of the faultline the earthquake happened

Table 6 - Explanation of Earthquake Class

User Interface:

Name	Type/Parameter	Return Type	Explanation
display	-	void	This function call provides user to open user interface to be filled
buttonHandler	-	void	This function call enables user to send query to get modelled result.
getModelledRequest	-	void	This function call is algorithmic side of the drawing path process of the request of mouse clicking

Table 7 - Explanation of User Interface Class

ArcGIS Connection:

Name	Type/Parameter	Return Type	Explanation

mouseClickHandler	-	void	This function activates the connecting to the database, sending the query to it and create a way map functionalities.
sendQuery	obj: District, obj: Earthquake	void	This function sends a query to the model in order to draw shortest path from the civil organizations to the most damaged areas in the selected district.
retrieveMap	obj: District	WayMap	This function returns the graph structured shortest path for the specific district in case of user request
connect	-	boolean	This function connects ArcGIS to the system database. It returns true for successful connection; false otherwise.
displayPath	-	void	This function displays different paths to the top three highest risk areas for nearest rescue agents.

displayRisk	-	void	This function displays a hotspot map according to user input by calculating collapse risk for all regions.
notify	-	void	This function captures the mouse click and notifies the class to use drawPath function

Table 8 - Explanation of ArcGIS Connection Class

MySQL Connection:

Name	Type/Parameter	Return Type	Explanation
connect	-	boolean	This function connects the Eclipse environment to the MySQL database.
createData	obj : Earthquake, obj : Building, obj : RescueAgent, obj : District, obj : Faultline, obj : WayMap	void	This function call creates a new entry in database considering all parameters, if one or more of the parameter is NULL, this parameter is assign to NULL variable.

readData	obj : Earthquake, obj : Building, obj : RescueAgent, obj : District, obj : Faultline, obj : WayMap	void	This function call retrieves an entry in database considering all parameters, if one or more of the parameter is NULL, this parameter matches will be ignored.
updateData	obj : Earthquake, obj : Building, obj : RescueAgent, obj : District, obj : Faultline, obj : WayMap	void	In database, current object's match will be established and parameters are updated regarding possible matches of arguments of the function.
deleteData	obj : Earthquake, obj : Building, obj : RescueAgent, obj : District, obj : Faultline, obj : WayMap	void	Matched database entry / entries will be deleted. If there is a NULL parameter, this will be ignored.
createModel	-	void	New risk map will be established in this function considering all data in database in the light of the created algorithm. Whole operations are performed in that function.

Table 9 - Explanation of MySQL Connection Class

The explanations of the class diagram given in Figure 5 are provided above. Getter and setter functions are ignored since they are not for functionality or algorithm of the system. They are the connectors of the private variables and the modifiers of them.

In explanation tables, since variables have no return type, they are indicated as '-'. On the other hand, function return types are written. For one Database there is only one ArcGIS connection. By the same approach, there can be only one user interface for the whole system. Hence, these are one-to-one relationship. In database, there may be several Building, District, RescueAgent, Earthquake objects. This is shown by 1 to many relationship which means for in MySQL Database Connection there can be several objects mentioned in before. This implies that database is data container of the Conveyor System. Other than this, if there exists a Faultline object it should belongs to Earthquake object/objects since faultline is associated with earthquakes. This results in a composition relationship between these two classes. Similarly, District and WayMap classes have same relationship. Since, the only chance of the WayMap object existence is depend on District object existence. In other words, if there is a way in somewhere it should belongs to a district. This is an another composition relationship. Other relationships are generalization relationships. Private variables are shown with '-' title. Public variables and functions are titled with '+'. ArcGISConnection, UserInterface and MySQLConnection classes are core classes of the Conveyor System.

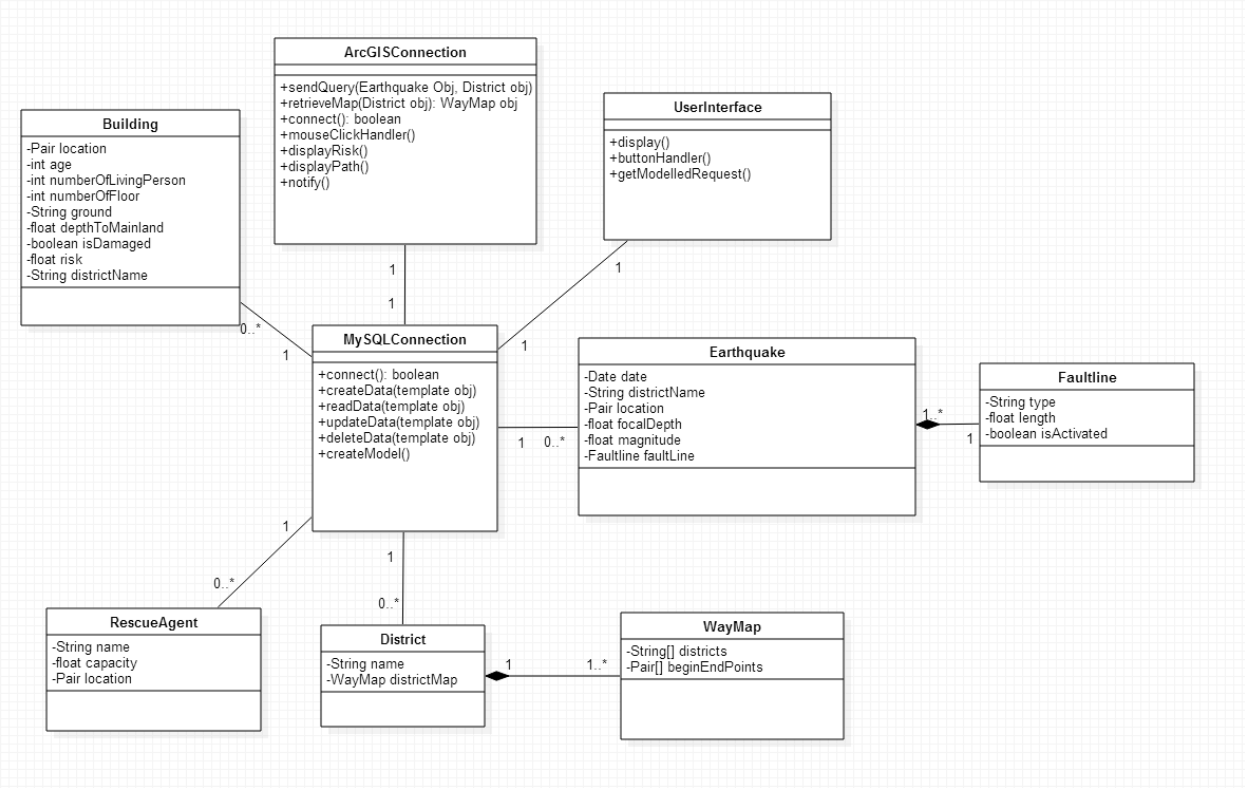


Figure 5 - Class Diagram of Conveyor

5.4.3. Example Languages

UML is used to form class diagrams by determining attributes and functions of it. Relationships between objects are explained in section 5.4.2.

5.5. Dependency Viewpoint

In this section, the association between user, desktop application and database will be examined. By the help of component diagram, (Please refer to section 5.6 for this component diagram) order of execution is visualised. Interconnection and access of elements are explained by stating parameterisation of interfaces and mutual information.

5.5.1. Design Concerns

The system composed of modules, services and interfaces. The connection between these

components is essential because they give an idea about system design which is very useful while investigating system failure, verification and validation process. Detailed package and component diagrams are explained in section 5.6 and functionalities of classes are explained in section 5.4.2 and they give idea about implementation level of the system. In general, system is highly dependent on database so database shall be filled up initially and accurately. User input interface needs a database to execute a query, ArcGIS interface shall be able to fetch data from database to display the output, algorithms shall be able to fetch appropriate data to process. Other than these, desktop application and database depends on algorithms because results of algorithms will be stored in the database and modeling will be done at desktop application by the help of algorithms.

5.5.2. Design Elements

Design entity: Algorithms

Graph traversal and risk calculation algorithms are held here. It depends on database which means it requires relevant data from the database. It provides well constructed model for desktop application entity.

Design entity: Database

Since the system is data dependent, this entity is vital for the product. It should contain all kind of information that algorithms require. Also it should accommodate output information. It depends on algorithms and desktop application, which means it requires relevant data from the algorithms to store the final output information and it requires generated data about buildings and other types of variables. It provides optimized output to visualization layer entity.

Design entity: Desktop Application

It consists of modeling and data generation part. It depends on Conveyor System User Interface, which means it requires relevant data from the user interface to pass that data to use it on modeling phase. As it is stated, it provides data to algorithms entity.

Design entity: Visualization Layer

Conveyor System User Interface package and ArcGIS interface are contained here. It depends on database which means it requires relevant output from the database to visualize.

Design entity: Conveyor System User Interface

User input interface can be found in this package. It provides appropriate data which client enters to desktop application.

5.5.2.1. Dependencies Attribute

All attributes for design entities are discussed in section 5.5.2.

5.5.3. Example Languages

UML is used while constructing the component diagram in section 5.6

5.6. Interface Viewpoint

This section provides interfaces of the software product to give an idea of model of the system. These are supported by component diagram to reify process. Hence, this viewpoint gives an idea to the designers, programmers and testers about correctness of the synthesis of services and design structure.

5.6.1. Design Concerns

Interface viewpoint remarks the interaction between interfaces, tools and libraries. The Conveyor System starts with user action. A user can start the system by filling the input areas of the provided user interface by relevant information explained in detail in SRS document and then submitting these information. After the intermediate calculations in modeling algorithm part ,which includes the determining possible risk by the help of the optimization algorithms, resulting values are shown to the user by the help of ArcGIS interface. ArcGIS fetches the relevant data including coordinates of the buildings and their risk rates from the database and creates a hotspot map. This hotspot map displays the interpolated risk percentages for a specific

area chosen by the client. After these calculations, user can click on the hotspot map to draw path according to the risk percentages of the inner area of the chosen district from nearest rescue agents. This path draw is also handled by developers.

In order to give more detailed explanation of the system, related Component diagram is provided in Figure 6.

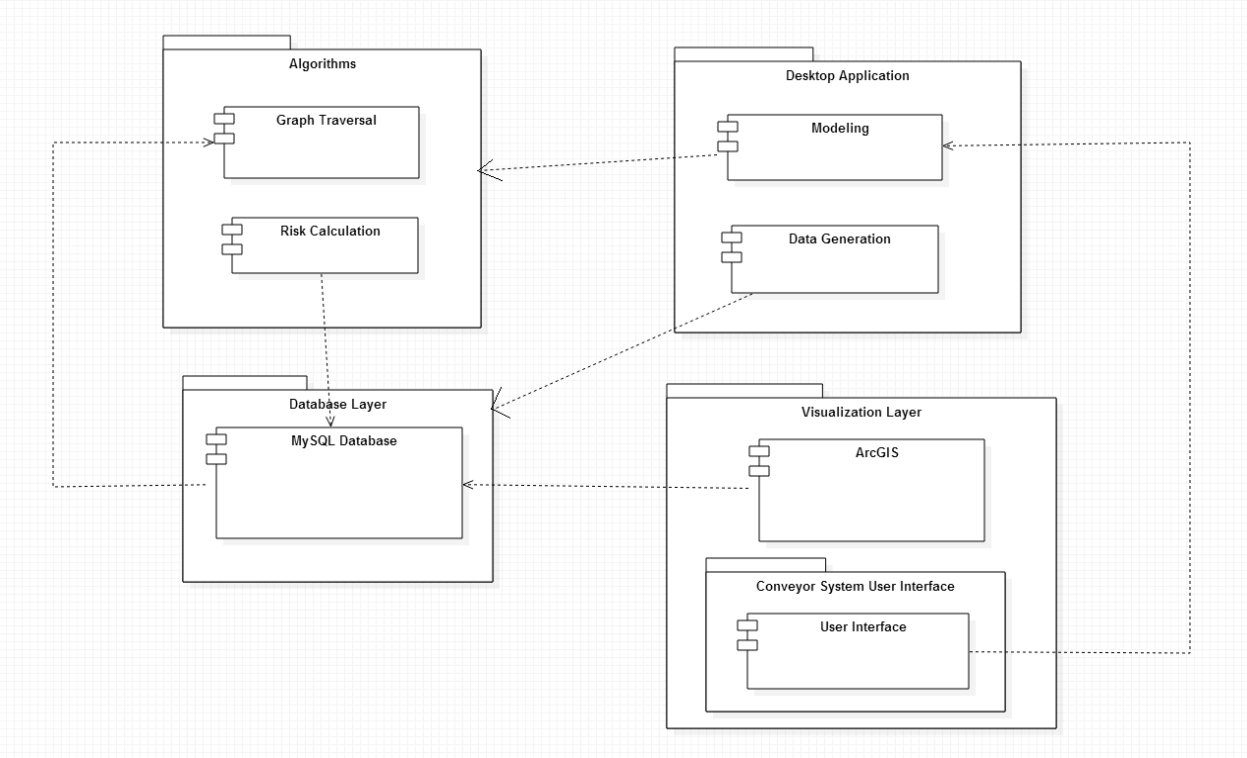


Figure 6 - Component Diagram of Conveyor

5.6.2. Design Elements

There are two different interfaces for Conveyor. User input interface is an user interface with four text boxes where each of which is associated with its own title (Latitude, Longitude, Magnitude, Focal Depth). It has one button. It is for sending earthquake information to the Conveyor system which is related with send query use case. Lastly there is one combobox. It allows users to choose a district for earthquake to be simulated. This whole interface is scalable which means user can maximize or minimize the interface window. In order to make interrogation, user should type all boxes with relevant information. Other than this, ArcGIS

interface is more different. Since ArcGIS is a private software, programmers will not design any interface for displaying. It has many buttons, combo boxes and options; however, Conveyor system will only use few of those options to display. In client perspective, clients will not be able to use any buttons of this interface, they will only visualize the output map. However, by mouse clicking user can change the way of the visualisation. This click will enable path draw mode and this mode invokes draw path use case in which the possible paths from the nearest rescue agent to the top 3 high risk areas are drawn. Developers will use interpolation and creating map options to provide an output map to clients. For route drawing, path algorithms are used. Scalability is same as user input interface and it is predetermined by ArcGIS developers. Both interface elements contain texts in English.

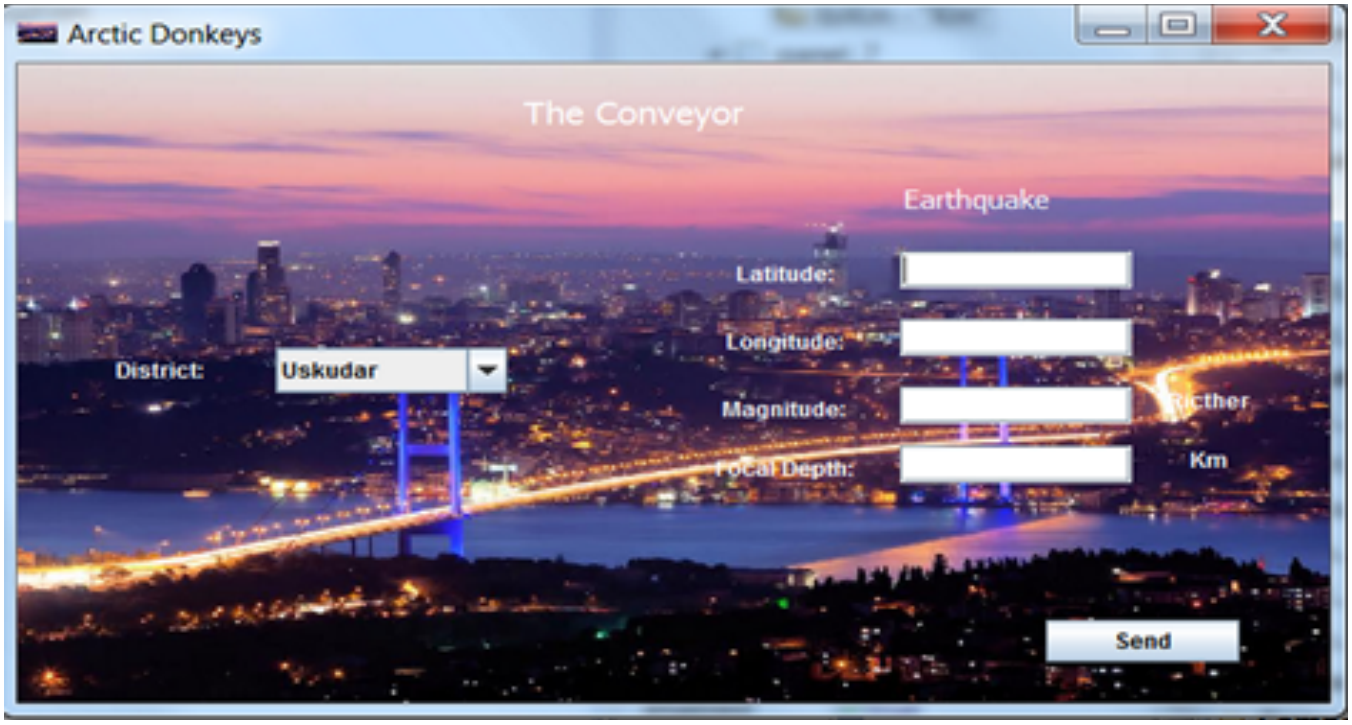


Figure 7 - User Input Page of Conveyor

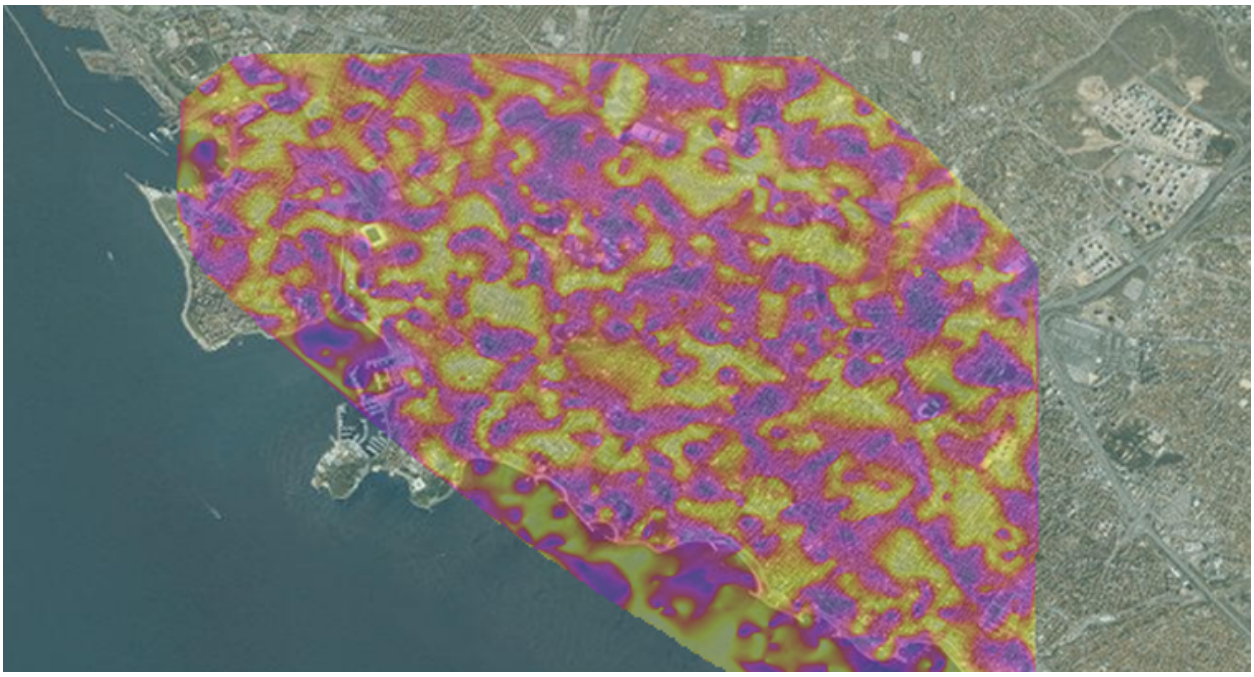


Figure 8 - ArcGIS Page Of Conveyor

5.6.3. Example Languages

Java Swing and Window Builder libraries are used to form the user input interface. UML is used while constructing the component diagram. For data dictionary, please refer to section 5.4.

5.7. Structure Viewpoint

In this section, internal elements and organization of the design theme will be discussed by using class, package and composite structure diagrams.

5.7.1. Design Concerns

Since related information about the class diagram is provided in section 5.4.2, and the package diagram is examined in section 5.3.2, these are not supplied again. Composite structure will be explained in this section.According to Spark Systems document, a composite structure diagram is a diagram that shows the internal structure of a classifier, including its interaction points to other parts of the system. It shows the configuration and relationship of parts, that together, perform the behavior of the containing classifier[6].

5.7.2. Design Elements

This section is merged with Composition, Logical and Dependency viewpoints. Please refer to sections 5.3, 5.4 and 5.5. In the composite structure diagram, behavior of desktop application entity is represented. It provides ArcGIS Interface and it requires data from Conveyor System User Interface. Desktop application entity is interacting with two environment entities which are algorithm and visualization entity. Also interfaces related with desktop application are detailed in the diagram. Methods of each interface can be observed.

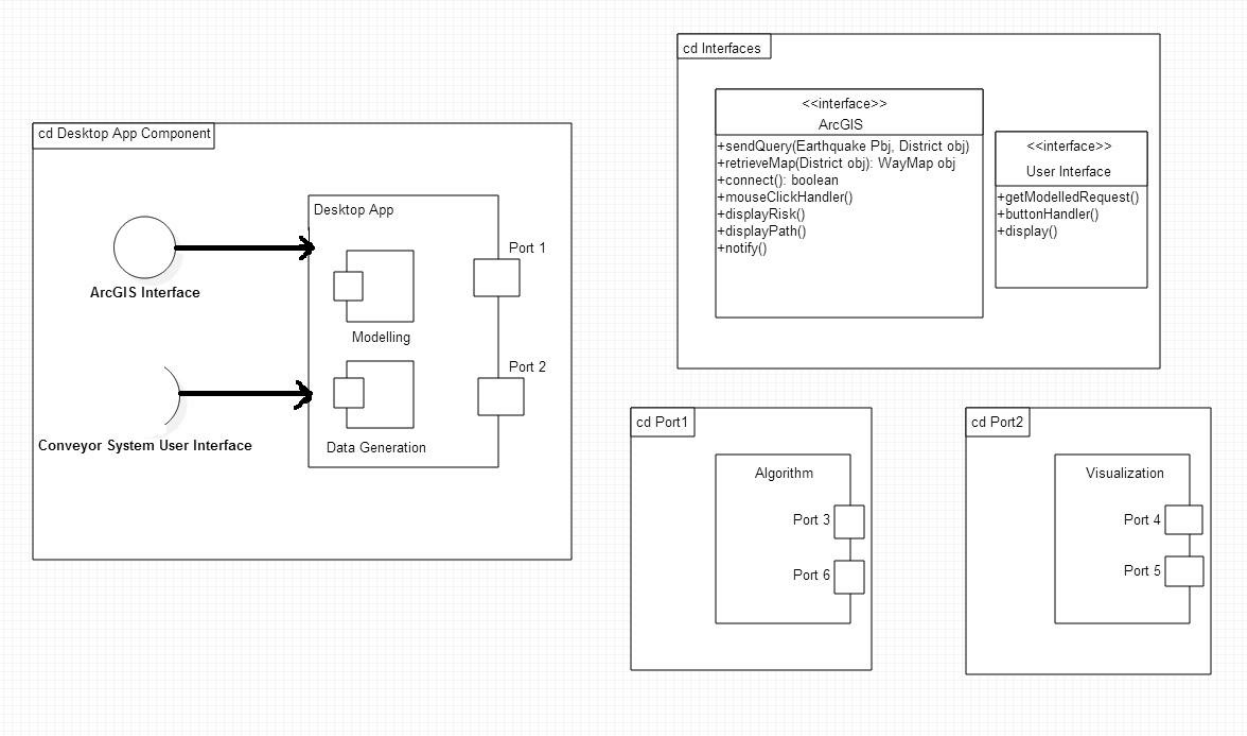


Figure 9 - Composite Structure Diagram of Conveyor

5.7.3. Example Languages

UML composite structure, class and package diagram is used to explain components of the Conveyor.

5.8. Interaction Viewpoint

In order to explain the flow of the functions by providing reasons, sequence diagrams will be used.

5.8.1. Design Concerns

The main goal is explaining dynamic structure of system providing run time behaviour.
Conveyor system represents distinct scenarios. They will be explained in 2 main sequence diagrams with different functionalities. The system mostly depends on user input and operations with these data. Moreover, functions should obey logical flow. For instance, user should provide legitimate input values so that former operations could be triggered.

5.8.2. Design Elements

First sequence diagram is about Client's use cases are illustrated in Figure 10. Client sends a query by filling the blanks of UserInterface. If relevant information comes from the user, buttonHandler() is activated and data goes to the Modelling part. Then, Java Eclipse connects to the MySQL DBMS to retrieve data to calculate risk. In case of successful connection, Model sends queries to retrieve data. In the light of retrieved information and input data, Model applies several test, optimization and modelling algorithms to the these data. This is done by getModelledRequest function. Then, by using modeling resulted risk and corresponding coordinates are used to update DBMS. After that, ArcGIS is notified. This notification awakes ArcGIS - MySQL connection and related function call is made. X-Y coordinates and their risks are retrieved from DBMS and this is displayed as hotspot map through ArcGIS. From this point, there are two choices. One is sending new query via UserInterface to the modelling and reprocessing it by the same approach. Other is awaking system by mouse clicking on the screen. This is done by mouseClickHandler function and makes modelling read data for drawing path process. Then, by displayPath function is called to display the path graph on the map and process is done.

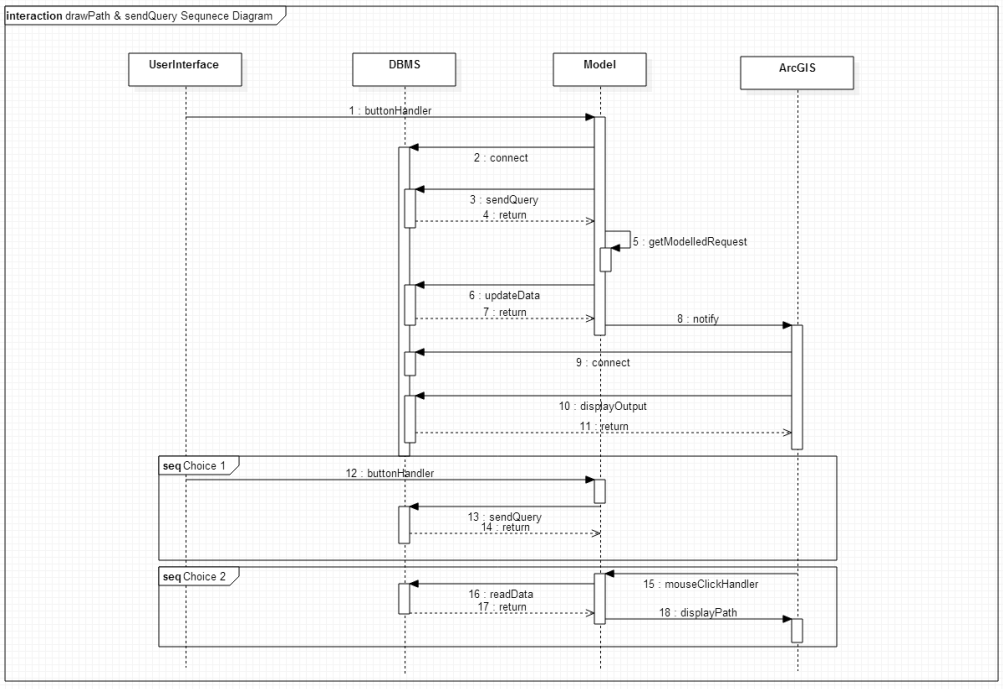


Figure 10 - Draw Path and Send Query Sequence Diagram of Conveyor

In second sequence diagram, Admin performs CRUD operations. In order to perform these operations, Admin needs to connect DBMS. Then, there are four different choices. Each one represents different schemas. One is representing add data, the other one is deleting data from DBMS. Similarly, update and read data are the different functionalities. Specified data is explained in Class diagram in section 5.4.

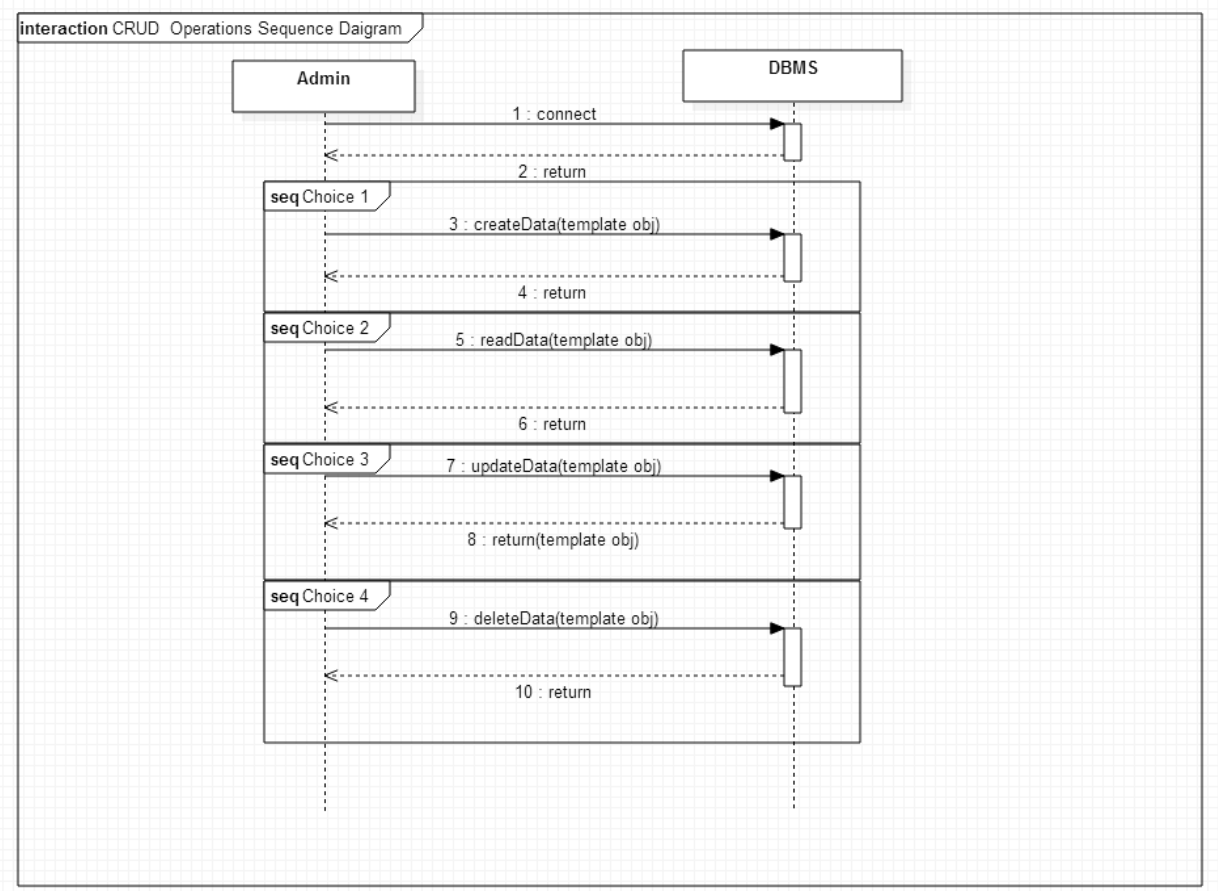


Figure 11 - Create Delete and Update Operation Sequence Diagram of Conveyor

Composite structure diagram is another diagram for representing dynamic behaviour of the system. For composite structure diagram and its detailed explanation, please refer to in section 5.7.2.

5.8.3. Example Languages

UML sequence diagram and composite structure diagram is used to explain dynamic behaviour of the Conveyor.

5.9. State Dynamics Viewpoint

This part of the SDD gives details about the reactivity and the internal behaviour of the Conveyor. Internal behavior of the Conveyor will be explained by using state diagram.

5.9.1. Design Concerns

The state diagrams show the behaviour of the Conveyor including modes, states, transitions and reactions to events. Transition of the flow is represented in each state. Stakeholders, developers and testers get dynamic structures of the Conveyor from the state diagrams.

The state chart in Figure 12 illustrates the behavior of the Conveyor System. In Conveyor System Admins creates data to train the system and these data are kept in MySQL database platform. After creation procedure admins can also read and update necessary data and delete irrelevant data to give the best result. Then, Admins use these data to train modeling system. All statistical calculations and regression testing mechanisms are performed at that stage. Then, system has a pattern about the earthquake properties-damage relationship. After that, system is ready for creating a model for incoming inputs. This model includes details about possible resultant destruction of the given earthquake with magnitude-latitude-longitude-focal depth information in chosen district. After all these processes, Client is able to use the system by sending queries by filling required data via user interface. This interface is created from Eclipse platform using Java Swing and Window Builder libraries . By default, ArcGIS-Eclipse and MySQL-Eclipse platforms were connected by Admins. ArcGIS and MySQL connection is established automatically. Given query is processed by trained data and outcome will give a model which estimates danger-input result. This result is passed ArcGIS through Eclipse as a processable data to be used in mapping procedure. ArcGIS displays the output as a hotspot map so that Client can simply understand the high risk areas and observe possible paths from the rescue agents to the prioritized locations. After this procedure, Client is also able to choose an area by clicking to see an optimized path considering nearest rescue agent. This mouse click activates reverse data flow from ArcGIS to Eclipse. Graph search algorithm is implemented in Eclipse and it gives the shortest path. This result is visualized through ArcGIS. Unless an action is performed, system preserves its current state.

5.9.2. Design Elements

After execution of the program, system enters a safe state. Conveyor System stays in Safe State in which user does not perform any action. That part divides into two other states. One of them is special for Admins including CRUD operations. In that point, Admins can perform one of the following operations; Create, Read, Update and Delete data. After performing one of the stated operations, a subsystem transition occurs which is Modification State. In that part user can either continue to perform database modification or to train Model using information contained in database. At that point, system enters Train Modified Data for Modeling state. This state applies statistical operations to whole data. After completion of the training process, Admins get the desired result and this state flow ends. Second alternative from the Safe State is that Client sends a query to the trained system. Client can keep sending queries without waiting for a result. After a successful querying operation, state switch occurs to Modelling Algorithm State. In that state, algorithm is applied using trained dataset to model the result. This result is sent to the related software and displayed on the ArcGIS interface. Unless a new action occurs , system keeps displaying the map. Other than this, there are three more options. One of them is the mouse click state where Client selects a zone from displayed map. If so, system shifts to Path Draw Algorithm state in which Graph Search Algorithm is performed. This algorithm draws shortest path and it is given to the Display Output on Map state. Displayed map is updated. Second option is closing the display map user interface and changes its state to the Send Query state. In that state, Client can send a new or same query by following the procedure. The last option is passing to end state where Client closes the software.

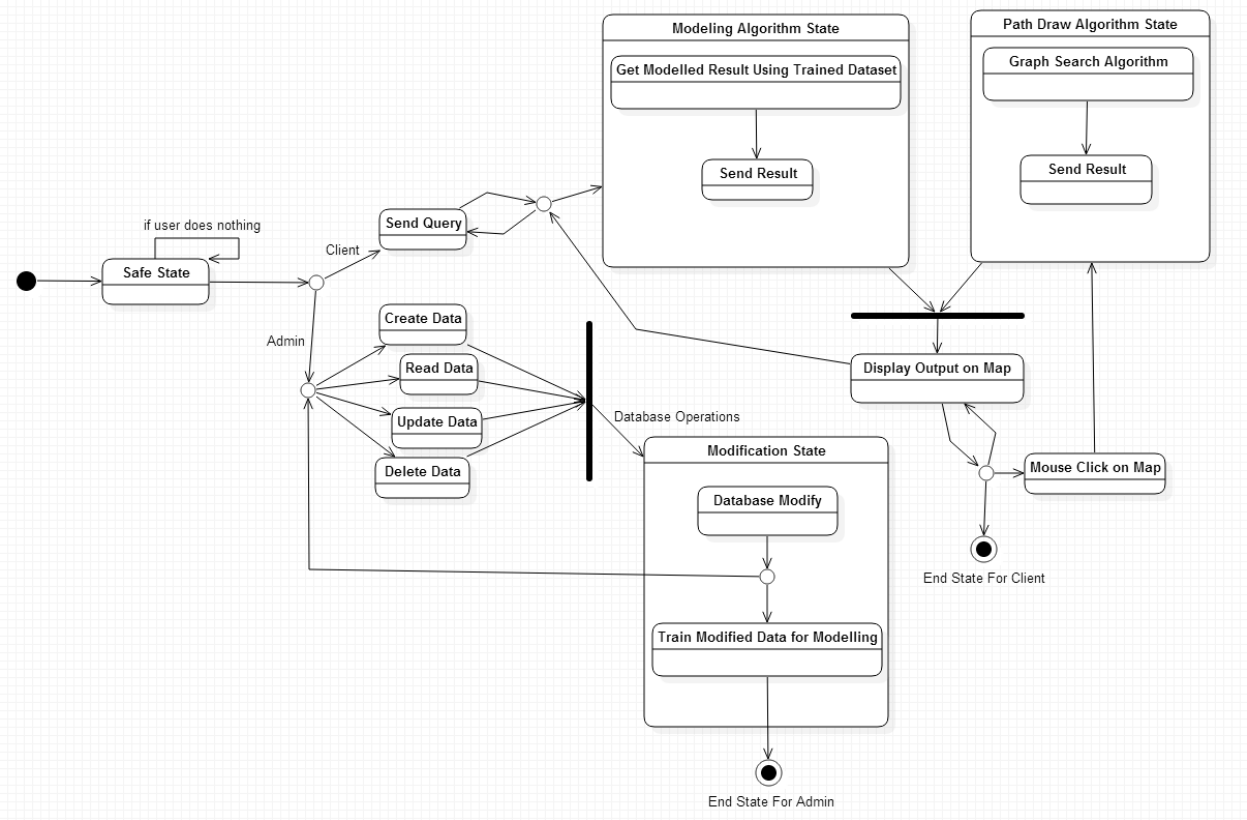


Figure 12 - State Diagram of Conveyor

5.9.3. Example Languages

In this section, dynamics of the system is represented by using UML state diagram.

5.10. Information Viewpoint

This section informs stakeholders about the logical structure of the software. Elaboration of entities and their relationships are discussed.

5.10.1. Design Concerns

Information viewpoint is highly used in development stages. It bothers with static structures of the system. Entities and relationships between them are explained through the section. Reuse concerns and adaptations are highly related with this viewpoint. In Figure 13, ER diagram is given. Explanations about ER diagram will be given in Design Elements (5.10.2)

section. Since the agile method is adopted, there may be slight changes in these diagrams. These changes will be updated swiftly in further versions.

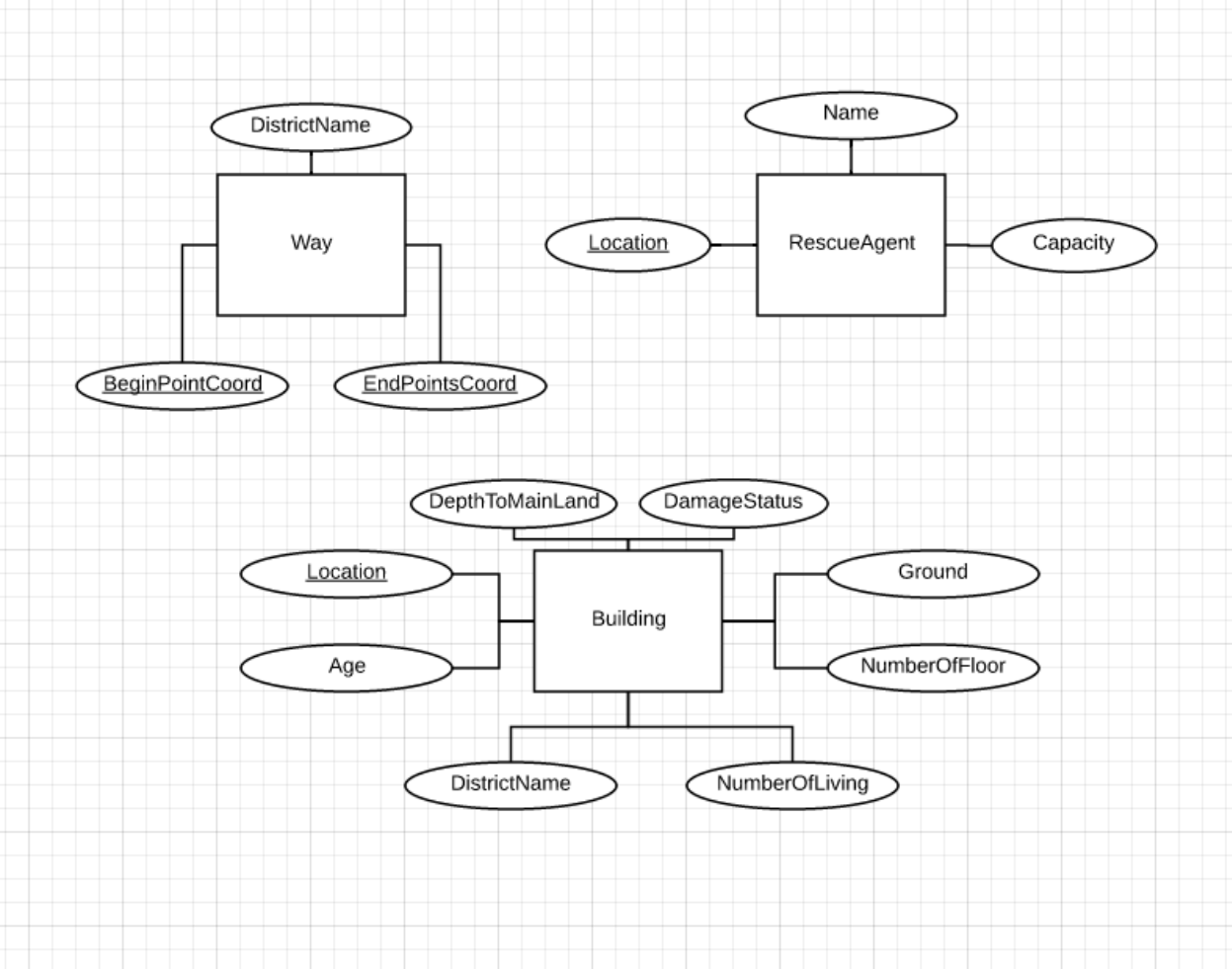


Figure 13 - ER Diagram of Conveyor

5.10.2. Design Elements

There are three entities in the ER diagram and relationships among them are illustrated.

RescueAgent: Entity for rescue agents. It has Location, Name, Capacity fields. Location is a unique identifier for rescue agents (Primary key). First field gives the information about the exact location of the rescue agent facility. Second field is the name of that specific rescue agent and the last field is about the support capacity of that rescue agent. It has no relationships with other entities.

Building: Entity for buildings. It has Location, Age, DistrictName, NumberOfLiving, NumberOfFloor, Ground, DamageStatus, DepthToMainLand fields. Location is a unique identifier for buildings (Primary key). First field gives the information about the exact location of the building. Second field is the age of that specific building. Third field carries information about the district where the building is located. Fourth field denotes the number of people living in that building. Fifth field denotes the floor number of that building. Sixth field is about the ground structure of the building. Seventh field states whether that building is damaged previously or not and the last field is about the distance from mainland to that building. It has no relationships with other entities.

Way: Entity for both district and waymap. It has BeginPointCoord, EndPointCoord and DistrictName fields. BeginPointCoord and EndPointCoord are composite primary key which is a unique identifier for way (Primary key). First field gives the information about the beginning coordinates of a street. Second field gives the information about the end coordinates of a street and the last field denotes the district where the street is located. It has no relationships with other entities.

5.10.3. Example Languages

UML is used while constructing the ER diagram.

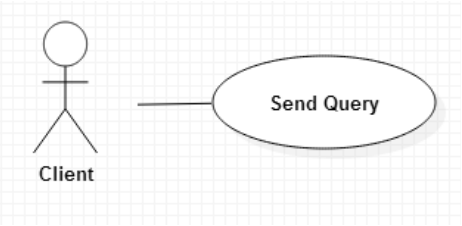
6. Conclusion

This Software Design Description Document provides particular information about the implementation phase of Conveyor. Supplying better attitude to stakeholders and testers is first aim of this document. Implementation of the Conveyor is mainly based on functional and nonfunctional requirements which are discussed in Software Requirement Specification Document. The implementation views are analyzed in terms of; context, composition, logical, dependency, interface structure, interaction and state dynamics viewpoints. User interfaces of Conveyor are also obtained from this document.

APPENDIX

Send Query Use Case:

Client sends query to the system. This query includes magnitude, latitude, longitude, focal depth and district information. System will try to find predicted outcome by comparing input and existing models.

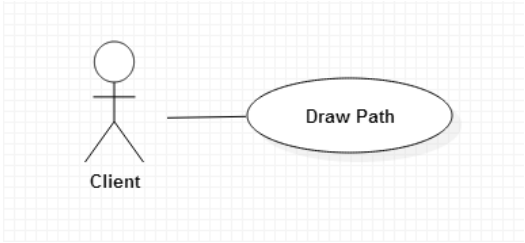


Use Case ID	Conveyor_UC1
Use Case Name	Send_Query
Description	Send_Query use case defines the user action that user can fill the criteria and send it to the system to get model on a map
Actors	Client
Preconditions	■ Client should open the Conveyor to make a query. ■ All empty text boxes should be filled ■ District area should be selected from dropdown list ■ All searching criteria should be filled with valid inputs which are discussed in section 2.1.2.
Trigger	This use case is activated when user fills all criteria and clicks "Send" button.
Basic Flow	Step 1: Client opens Conveyor system Step 2: Client fills all query criteria by considering valid input ranges. Step 3: Client presses "Send" button. Step 4: Conveyor models the given query using training data. Step 5: Result model is illustrated via ArcGIS interface.
Alternate Flow	There is no alternate flow this use case.

Exception Flow	■ If Client does not select any district and tries to send the query, system gives “Please select a valid district area” error. ■ If Client does not fill Latitude box or entered Latitude value is out of range, system gives “Valid entry should be between 40.28-41.33” message. ■ If Client does not fill Longitude box or entered Longitude value is out of range, system gives “Valid entry should be between 28.01-29.55” message. ■ If Client does not fill Magnitude box or entered value is not valid, system gives “Valid entry should be between 4-10” message. ■ If Client does not fill Focal Depth box or entered value is not valid, system gives “Valid entry should be between 0-200” message. ■ If multiple areas are not filled and/or filled with invalid data error message is given regarding upper left located entry.

Draw Path Use Case:

If a user wants to draw a path to the zone s/he wants, s/he shall click to the appropriate zone on the output map provided by ArcGIS. Then, the shortest path from the nearest rescue agent to that zone will be drawn.

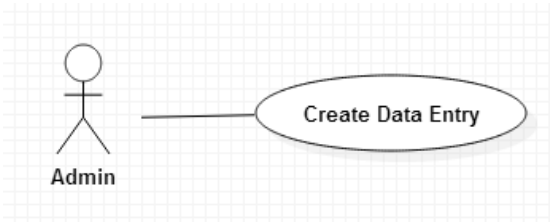


Use Case ID	Conveyor_UC2
Use Case Name	Draw_Path
Description	Draw_Path use case defines the user action that user can select a special area on map to see shortest path to the that destination from the nearerst rescue agent.
Actors	Client

Preconditions	In order to do action on created map, Client should send query to the Conveyor to create a map.
Trigger	Mouse clicking on map
Basic Flow	Step 1: Client selects a desired area on map by mouse click. Step 2: Conveyor handles this request and uses Graph Search Algorithm to draw path from the nearest rescue agent to the desired location Step 3: Result path is visualised through ArcGIS
Alternate Flow	There is no alternate flow for this action
Exception Flow	Since map is not created from the Conveyor unless a valid query search performed, there is no exceptional flow.

Create Data Use Case:

Admin creates data for whole optimization and visualisation process. Main algorithm uses created entries to produce a trained model.

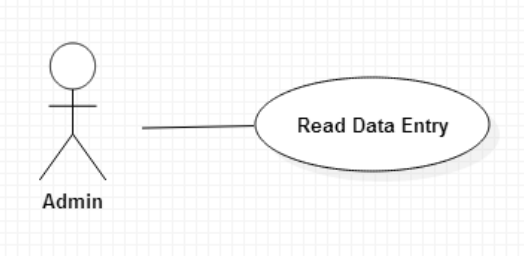


Use Case ID	Conveyor_UC3
Use Case Name	Create_Data_Entry
Description	Create_Data_Entry use case defines the user action that user can fill the entities and add it to the system to create data for training action.
Actors	Admin
Preconditions	Connecting the database shall be established
Trigger	Since these actions are handled by java functions automatically, there is no special trigger button for it.

Basic Flow	Step 1: Admin connects database Step 2: Admin fills entries to add them to the database Step 3: By function call, data are added to database.
Alternate Flow	There is no alternate flow.
Exception Flow	If there is any insufficient data entity, Java functions throw exception which is handled by Admin.

Read Data Use Case:

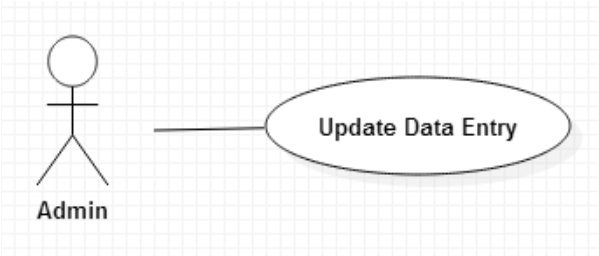
Admin can fetch data from database to observe and process data.



Use Case ID	Conveyor_UC4
Use Case Name	Read_Data_Entry
Description	Read_Data_Entry use case defines the user action that user can fetch the entities and investigate them for any errors.
Actors	Admin
Preconditions	Connecting the database shall be established
Trigger	Since these actions are handled by java functions automatically, there is no special trigger button for it.
Basic Flow	Step 1: Admin connects database Step 2: Admin fetches entries to investigate them
Alternate Flow	There is no alternate flow.
Exception Flow	If the database does not contain the queried data, Java functions throw exception which is handled by Admin.

Update Data Use Case:

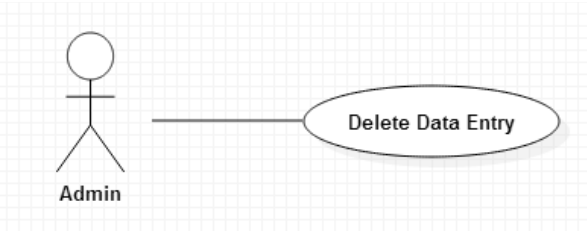
Admin can update the data in the database.



Use Case ID	Conveyor_UC5
Use Case Name	Update_Data_Entry
Description	Update_Data_Entry use case defines the user action that user can change the entities in database
Actors	Admin
Preconditions	Connecting the database shall be established
Trigger	Since these actions are handled by java functions automatically, there is no special trigger button for it.
Basic Flow	Step 1: Admin connects database Step 2: By function call, updates are reflected to the database.
Alternate Flow	There is no alternate flow.
Exception Flow	If the query to the database is erroneous, Java functions throw exception which is handled by Admin.

Delete Data Use Case:

Admin can remove a specific data from the database.



Use Case ID	Conveyor_UC6
Use Case Name	Delete_Data_Entry
Description	Delete_Data_Entry use case defines the user action that user can remove the entities for changes in previous data or renewed data.
Actors	Admin
Preconditions	Connecting the database shall be established
Trigger	Since these actions are handled by java functions automatically, there is no special trigger button for it.
Basic Flow	Step 1: Admin connects database Step 2: Admin specifies entries to detele them Step 3: Changes are reflected through database.
Alternate Flow	There is no alternate flow.
Exception Flow	If the query to the database is erroneous, Java functions throw exception which is handled by Admin.

INDEX