**FLOOD serv**

**{flood risk management}**

-akanksha kumari

**Introduction**

A disaster is a consequence of a sudden disastrous event which seriously disrupts the normal function of the society or the community to the extent that it cannot subsist without outside help. A disaster is not just the occurrence of an event such as an earthquake, flood, conflict, health epidemic or an industrial accident; a disaster occurs if that event/process negatively impacts human populations. *Disasters combine two elements: hazard, and the vulnerability of affected people.* A disaster occurs when a hazard exposes the vulnerability of individuals and communities in such a way that their lives are directly threatened or sufficient harm has been done to their community's economic and social structure to undermine their ability to survive. A disaster can be defined as any tragic event stemming from events such as earthquakes, floods, catastrophic accidents, fires, or explosions.

Disaster response is the second phase of the disaster Management cycle. It consists of a number of elements, for example, warning, evacuation, search and rescue, providing immediate assistance, assessing damage, continuing assistance and the immediate restoration. So among all, I have worked upon warning system for floods. In this, we have provided a user interface to the common public to check the level of water flow in rivers in future and have a provided a mechanism of notification if there is any possibility of flood due to any river in nearby future(12 months). Along with that users can also see the historical trends of rivers flow and can visualize the rainfall patterns also in their Sub-Division (Area). So with that much information beforehand and knowing the chances of the flood in any region we can prepare ourselves and alert the local public so that loss would be minimum.

# Objective

The overall objective of FLOOD-serv is **to provide a pro-active and personalized citizen-centric public service application** that will encourage citizens’ involvement and will the involvement of the citizen and will harness the collaborative power of ICT[information and communication technology] networks (networks of people, knowledge and sensors) to raise awareness on flood risks and enable collective risk mitigation solutions and response actions.

The **general objectives of FLOOD-serv** are:

1. *To empower local communities to directly participate in the design of emergency services dealing with mitigation actions for floods.*
2. *To harness the power of new technologies, such as social media and mobile technologies, to increase the efficiency of public administrations in raising public awareness and education regarding floods risks, effects and impact.*
3. *To encourage the development and implementation of long-term, cost-effective and environmentally sound mitigation actions related to floods through an ICT-enabled cooperation and collaboration of all stakeholders: government, private sector, NGOs and other civil society organizations as well as citizens.*

The **specific objectives of FLOOD-serv** are:

1. *to make use of the best available data in order to identify the location and potential impacts that natural hazards as floods can have on people, property and natural environment*
2. *to improve the systems of warning and emergency communications*

*3. to provide support for the public authorities and government institutions’ hazard mitigation efforts, including planning and action coordination*

*4. to inform the public on the risk exposure to natural hazards and how they can get prepared, respond, recover and mitigate the impacts of such events*

So given the above mentioned objectives, the project will offer an opportunity for collective problem solving, knowledge sharing, social exchange and community-wide participation at local and global scale. This will lead to an insight into the information and preparedness requirements of local communities and the development of solutions adapted to the social realities.

Secondly, it will lead to a closer cooperation and coordination for flood forecasting and warning services of public institutions based on user needs.

Thirdly, based on the flood event studies, and including consultations with affected communities and other recipients of flood warnings, improved technical means of detecting the areas at imminent risk and warning more effectively, will be developed.

Technically the project will focus on developing a collaborative platform that will link citizen, public authorities and other stakeholders and on enabling the public to be warned en masse so that actions can be taken to reduce the adverse effects of the flood.

# Scope

The Flood serv set out a sustainable, long-term strategy to manage the flood risk within each River Basin, focused on the areas of potentially significant flood risk and the sources of flooding giving rise to that risk.

The scope of each Flood Risk Management Plan is:

The Flood Serv set out the range of policies and measures, which are in place, under development or proposed that contribute to the reduction and management of flood risk throughout the River Basin.   
  
The Plans also set out the viable measures, typically flood relief schemes, proposed to manage and reduce the flood risk in the communities that were identified as being at potentially significant flood risk.

The measures set out in the Flood Plans address the sources of flooding identified as potentially significant in one or more communities within the area covered by each River Basin Plan.

The Flood Plans set out the measures that are proposed as the most appropriate at this stage of assessment.

# Module

water level  
flood prone areas  
weather history  
rescue shelter  
policy and plan  
contact   
river basin  
tragedy till now  
animals  
donation corner  
inventive ideas: developer and thinkers

# Project Category

A database system is essentially a sophisticated, computerized record keeping system, a repository for a collection of computerized data files. A database system maintains information and makes that information available on demand, for this purpose a database system provides set of facilities to perform such operations.

The most important advantage of the database is to maintain the integrity i.e. It insures that the changes made to the database by authorized users do not result in a loss of data consistency and guard against user’s damage to the database.

**RDBMS have the following facilities:**

 Creation of files, Addition of data, Deletion of data, Modification of data.

 Retrieving the data collectively or selectively.

 The data stored can be sorted or indexed at users discretion or direction.

 Various reports can be produced from the system. These may either be standardized reports or that may be specifically generated according to specific user definition.

 Mathematical function can be performed and the data stored in the database can be manipulated with functions to perform the desired calculations.

 To maintain data integrity and database use.

 Data integrity for multiple users.

**OOPS**

 Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which are data structures that contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods. A distinguishing feature of objects is that an object's procedures can access and often modify the data fields of the object with which they are associated (objects have a notion of "this"). In object-oriented programming, computer programs are designed by making them out of objects that interact with one another. There is significant diversity in object-oriented programming, but most popular languages are class-based, meaning that objects are instances of classes, which typically also determines their type.

 Many of the most widely used programming languages are multi-paradigm programming languages that support object-oriented programming to a greater or lesser degree, typically in combination with imperative, procedural programming. Significant object-oriented languages include C++, Objective-C, Small-talk, Delphi, Java, Python, and PHP.

**Hardware & Software, Tools/Platform Requirement Specification**

**Software requirement:**

**Operating System:** Window XP, 2003, 2007

**Platform:** Visual Studio Code

**Tools:** MySQL

**Front-End:** HTML5, CSS3, Bootstrap, JavaScript

**Back-End**: PHP

**Server:** Apache Server

**Browser:** Internet Explorer, Google Chrome, Mozilla Firefox etc.

**Hardware Requirement:**

**Monitor:** VDU

**Processor:** P4 Processor having (550 to 933 MHZ)

**RAM:** 128 MB RAM

**HDD**: 20 GB HD CD-ROM

**DRIVE:** 52X

**MODEM:** 36.6 KBPS

**PRINTER:** DeskJet

**Software Model**

**HTML**

**HTML** stands for **Hypertext Markup Language**. It allows the user to create and structure sections, paragraphs, headings, links, and block-quotes for web pages and applications.

HTML is not a programming language, meaning it doesn’t have the ability to create dynamic functionality. Instead, it makes it possible to organize and format documents, similarly to Microsoft Word.

When working with HTML, we use simple code structures (tags and attributes) to mark up a website page. For example, we can create a paragraph by placing the enclosed text within a starting ***<p>*** and closing ***</p>*** tag.

Overall, HTML is a markup language that is really straightforward and easy to learn even for complete beginners in website building.

# CSS

**C**ascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

**Advantages:**

* **CSS saves time** − You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
* **Pages load faster** − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
* **Easy maintenance** − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
* **Superior styles to HTML** − CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
* **Multiple Device Compatibility** − Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
* **Global web standards** − Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

# Bootstrap

* Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website.
* It is absolutely free to download and use.
* It is a front-end framework used for easier and faster web development.
* It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others.
* It can also use JavaScript plug-ins.
* It facilitates you to create responsive designs.

**The main advantage of Bootstrap:**

* It is very easy to use. Anybody having basic knowledge of HTML and CSS can use Bootstrap.
* It facilitates users to develop a responsive website.
* It is compatible on most of browsers like Chrome, Firefox, Internet Explorer, Safari and Opera etc.

# JavaScript

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as **LiveScript,** but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name **LiveScript**. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

* JavaScript is a lightweight, interpreted programming language.
* Designed for creating network-centric applications.
* Complementary to and integrated with Java.
* Complementary to and integrated with HTML.
* Open and cross-platform

# Client-Side JavaScript

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.

The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.

JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

# Advantages of JavaScript

The merits of using JavaScript are −

* **Less server interaction** − You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
* **Immediate feedback to the visitors** − They don't have to wait for a page reload to see if they have forgotten to enter something.
* **Increased interactivity** − You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
* **Richer interfaces** − You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

# Limitations of JavaScript

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features −

* Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
* JavaScript cannot be used for networking applications because there is no such support available.
* JavaScript doesn't have any multi-threading or multiprocessor capabilities.

Once again, JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

# PHP

* PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.
* PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
* PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
* It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
* PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
* PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
* PHP is forgiving: PHP language tries to be as forgiving as possible.
* PHP Syntax is C-Like.

# Common uses of PHP

* PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
* PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.
* You add, delete, modify elements within your database through PHP.
* Access cookies variables and set cookies.
* Using PHP, you can restrict users to access some pages of your website.
* It can encrypt data.

# Characteristics of PHP

Five important characteristics make PHP's practical nature possible −

* Simplicity
* Efficiency
* Security
* Flexibility
* Familiarity

**SRS Document**

It is a reference document or contract between the customer and the development team. Once the customer agrees to the SRS document, the development team proceeds to develop the product conforming to all the requirements mentioned in the SRS document.

An SRS document should clearly document the following:

1. **Functional requirements of the system:** Each function of the system can be considered as performing a transformation of a set of input data to the corresponding set of output data. The functional requirements of the system should clearly describe each of the functions that the system needs to perform along with the corresponding input and output data set.
2. **Non-functional requirements of the system:** Non-functional requirements deal with the characteristics of the system that cannot be expressed functionally, e.g., maintainability, portability, Usability, etc. The non-functional requirements also include reliability issues, accuracy of results, human computer interface issues, operating and Physical constraints, etc.
3. **Constraints on the system:** The constraints on the use of the system may describe certain things that the system should or should not do.

**Nature of SRS**

The basic issues the SRS writer(s) shall address are the following:

1. **Functionality:** What the software is supposed to do?

2. **External Interfaces:** How does the software interact with people, the system’s hardware other hardware and other software.

3. **Performance:** What is the speed, availability, response time, recovery time, etc., of the various software fundamentals.

4. **Attributes:** What are the consideration for portability, correctness, maintainability, security, reliability, etc.

5. **Design constraints imposed on an implementation:** Are there any required standards or effect, implementation language, policies for database, integrity resource limits, operating environment, etc.

**Characteristics of a good SRS**

An SRS should be

Correct

Unambiguous

Complete

Consistent

Ranked for Importance and for Stability

Verifiable

Modifiable

Traceable

**Correct:** There is no tool or procedure that assures correctness. If the software must respond to all button presses within 5 seconds and the SRS stated that “the software shall respond to all button presses within 10 seconds”, then that requirement is incorrect.

**Unambiguous:** An SRS is unambiguous if and only if every requirement started therein has only are interpretation. In cases, where a term used in a particular context could have multiple meanings, the term should be included in a glossary where its meaning is made more specific.

**Complete:** An SRS is complete if and only if it includes of the following elements.

1. All significant requirements, whether relating to functionality, performance, design constraints, attributes or external interfaces.

2. Full labels and references to all figures, tables and diagram in the SRS and definition of all terms and units of measure.

**Consistent:** An SRS is consistent if no subset of individual requirements desorbed in it conflict. There are 3 types of likely conflicts in an SRS:

1. The specified characteristics of real word objects may conflict, e.g.

a. The Format of an output report may be described in are requirements as tabular but in another as textual.

b. One requirement may state that all lights shall be green while another states that all lights should be blue.

2. There may be logical or temporal conflict between two specified actions, e.g.,

a. Are requirement may specify that the program will add 2 inputs and another may specify that the program will multiply them.

b. Are requirement may state that ‘A’ must always follows B, while another requires that A&B occur simultaneously.

3. Two or more requirements may describe the same real word object but use different terms for that object. The use of standard terminology and definitions promotes consistency.

**Water Fall Model**

The **waterfall model** is a sequential design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall model) through the phases of Conception, Initiation, analysis, design, Construction, testing.

Production/implementation and maintenance.

The waterfall development model originates in the manufacturing and construction industries; highly structured physical environments in which after-the-fact changes are prohibitively costly, if not impossible. Since no formal software development methodologies existed at the time, this hardware-oriented model was simply adapted for software development.

This presentation was about the development of software for SAGE.



# PROJECT SCHEDULING

**GANTT CHART**

* It is used, for scheduling technique. Gantt chart can be developed for the entries project.
* A tabular representation is maintained where row indicate the table or modules and column
* indicate duration in week or month.
* Circle is use used to represent milestone or deliverable.
* Rectangle is used to represent duration of completion of the table.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | August | August |  |  |  |  |  |  |
| Tasks | 8 | 20 | 24 | 30 | 10 | 30 |  |  |
| Assign Teams |  |  |  |  |  |  |  |  |
| Select Reverse Engr. Project |  |  |  |  |  |  |  |  |
| Write prosposal |  |  |  |  |  |  |  |  |
| Make charts and Diagram |  |  |  |  |  |  |  |  |
| Writing Final Report |  |  |  |  |  |  |  |  |

# DataFlow Diagram

A DFD maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles, and arrows plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled.

Customer

DATABASE

ADMIN

**0 Level DFD**

Update, Register

LogIn, Provide details

Complaint, Feedback,

Give local information

Receive

information

blank