



Intelligent Notice Board

CSP315 Sem I , 2010-2011

Abhinav Punia 2008CS10153
Ankit Tomar 2008CS10158
Anupam Dev Goel 2008CS10163
Sahil Aggarwal 2008CS10187
Swapnil Jain 2008CS10194



INDEX

1. Specifications	3
2. Block Diagram	7
3. Methodology	8
4. Major Components/Kits	9
5. Time line and division of work	10
6. Web link of project	14

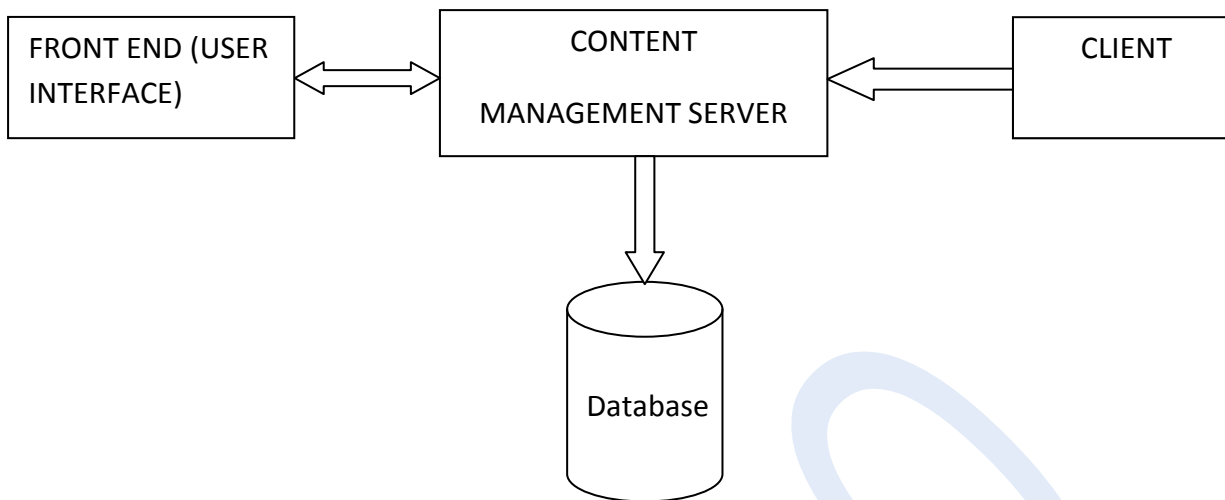
MMMOBO

Specifications

1. CLIENT

- Provision of different screen layouts to support varying needs.
Till now there has been provided only a single layout, which is a very rigid approach, because same layout is not required at every place. E.g. the hostel screens may not require Tickers. So each client will be provided with sample layouts for display which can be changed if and when needed but only by administrator.
- Improving quality of display.
We will be aiming at the improvement of image display so that the images change smoothly as they are not doing now.
- Displaying text directly on the screen (Rather than Image currently used to display text)
Till now in Tickers and sidetapes, the user enters the text and an image is formed which is displayed. In this process a lot of bandwidth is wasted as images are heavier than text. So our goal will be to directly upload the text instead of first forming an image.
- Modularity and Robustness in the code.
The current code modules are not fully robust as sometimes some problems crop up. So the emphasis will be on handling those problems.
- Integration with 3G for communication.
At many places there are no wired or WiFi networks available, whereas the 3G networks can be availed at any place and without much fuss. Thus we will be trying our hands at integrating the hardware with the new 3G technology for communication.
- Development of new custom hardware (PCB).
Till now we have been working on DM355. The available hardwares contain some redundant devices which we do not require but they add up to the cost, e.g. audio codec. So making our own custom hardware will be a major goal of the whole process.

2. SERVER



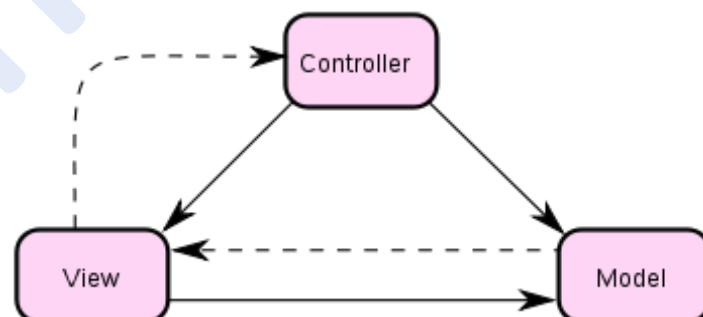
The Server Side will essentially be based on the **MVC (MODEL – VIEW – CONTROLLER)** software architecture pattern.

This pattern isolates "domain logic" (the application logic for the user) from input and presentation (UI), permitting independent development, testing and maintenance of each.

The **model** is used to manage information (Data) and notify observers when that information changes.

The **view** renders the model into a form suitable for interaction, typically a user interface element.

The **controller** receives input and initiates a response by making calls on model objects.



Model-View-Controller concept. The solid line represents a direct association, the dashed an indirect association(via an observer for example). Resource :- Wikipedia .

✓ **MODEL**

- **DATA Content Management Server will Manage**

- 1) Posters
- 2) Videos
- 3) Text/RSS
 - a) Ticker Tape
 - b) Side Tape
- 4) Information About Client Screen
- 5) Users

- **INFORMATION THAT WILL BE STORED FOR POSTERS/TICKER/SIDETAPE/VIDEOS**

- 1) Path the file to be uploaded or text in case of ticker and sidetape.
- 2) Final Resolution
- 3) User Reference
- 4) Date Uploaded
- 5) Start Date
- 6) End Date
- 7) Precedence / Order of posters
- 8) Display Time
- 9) Date of Event
- 10) Client Screens on which poster/ticker/sidetape are to be displayed
- 11) Description/Title

✓ **VIEW**

- User Interface (UI)
- Administrator Interface
- Poster Making Interface

✓ **CONTROLLERS**

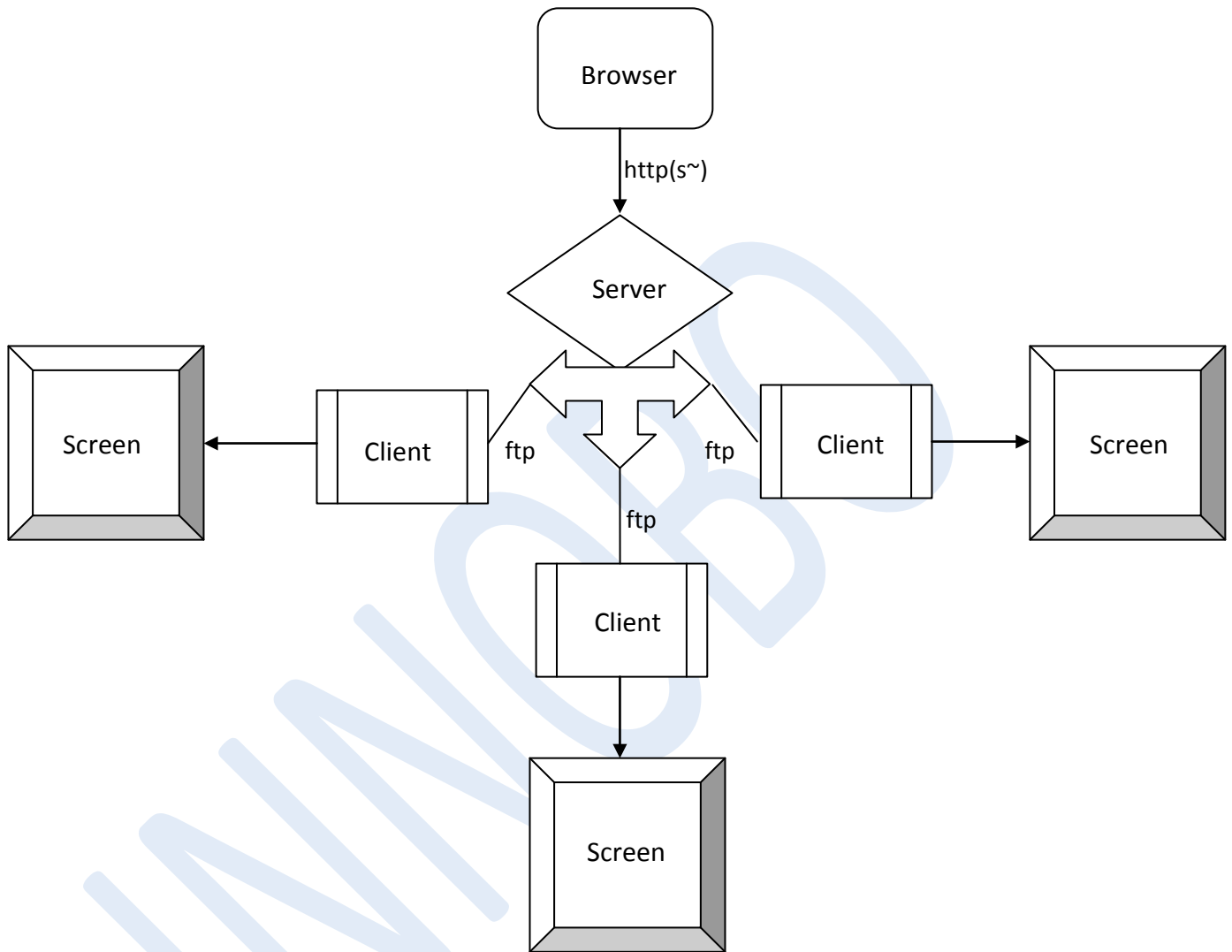
- File Handler
- Login Control (Authentication and Moderation with LDAP)

- Poster Making
- RSS update
- Status Controllers

Final improvements over the current system:

- Authentication & Moderation using LDAP
- New User Interface which handles and improves all the previous shortcomings and errors.
- Robustness in the code.
- Status of screen – power issues, board functioning.
- Poster making software on website.
- Creating more administrative options like control over user privileges.
- Controlling poster sequence
- Scalability

Block Diagram



Methodology

PULL MECHANISM

The server collects all the poster, side tapes and ticker tapes from the user. The client pings the server repeatedly after certain interval of time. And gets the uploaded files themselves and all the necessary files containing data regarding the uploads. Then it checks the expiry of the uploads and send them to the buffer for display, if they are not expired. It also keeps track of the delete requests made, if any.

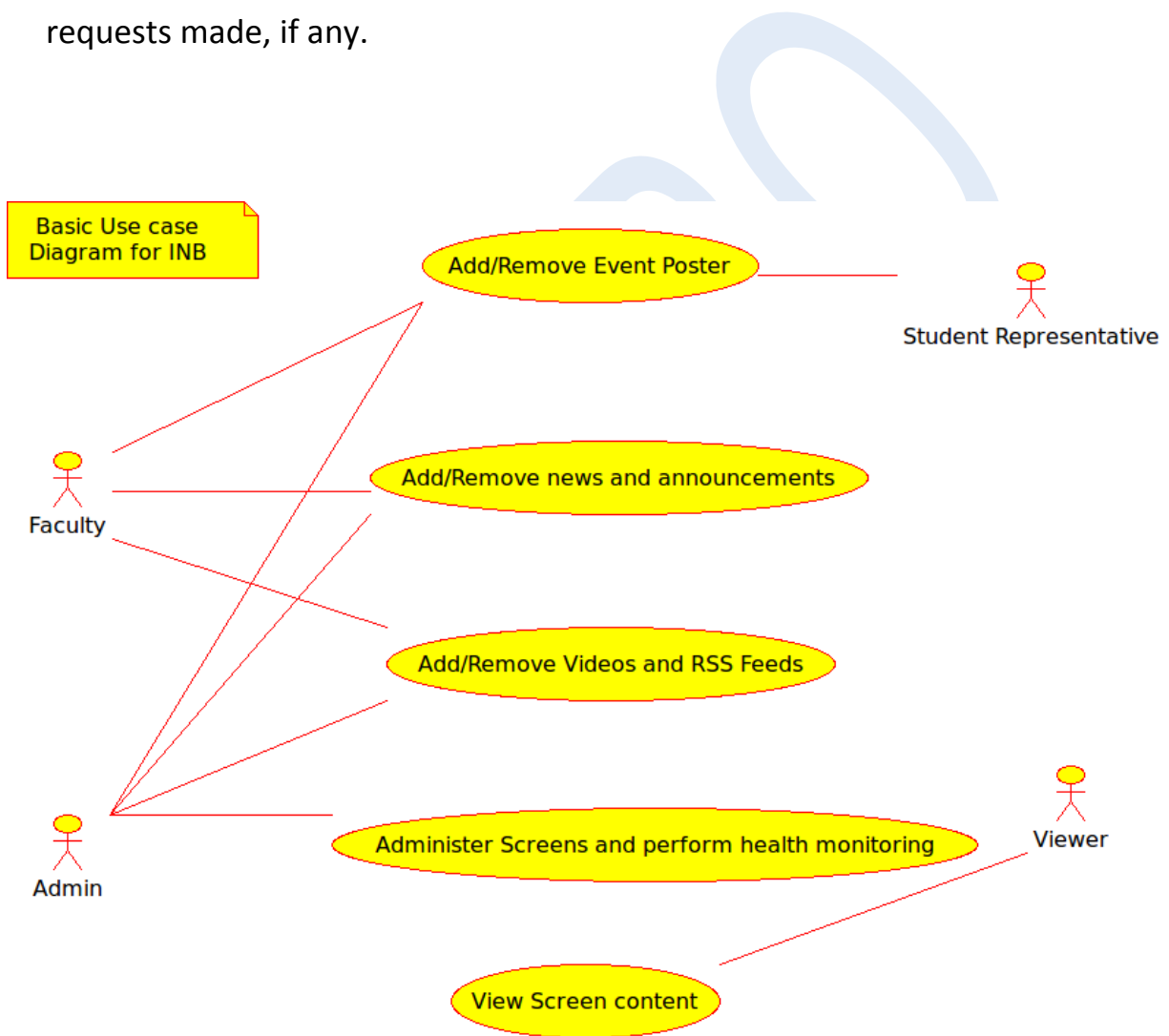


Fig 1: Use Case Diagram

Major Components/Kits

1. Till now we have been working with Leopard Board 355. Now we are planning to shift to Leopard Board 365 based on TMS320DM365 processor for the following reasons:

- Better quality of pictures
- Video support

Details and capabilities of the board are:

- ARM926EJ-S Core: 216, 270 and 300 MHz
- Video Processing Subsystem
- Two video image co-processors (HDVICP, MJCP) engines
- Hardware On-Screen Display(OSD)
- HD Video Codecs: H.264, MPEG4, MPEG, WMV9/VC1, MPEG2
- SD memory card support
- Composite TV/Component HD TV output
- 10/100 Ethernet Port
- Capable of 720p 30fps H.264 video processing

2. LCD Screen

3. SD card

4. Component to support connection with server via 3G

Time Line and division of work

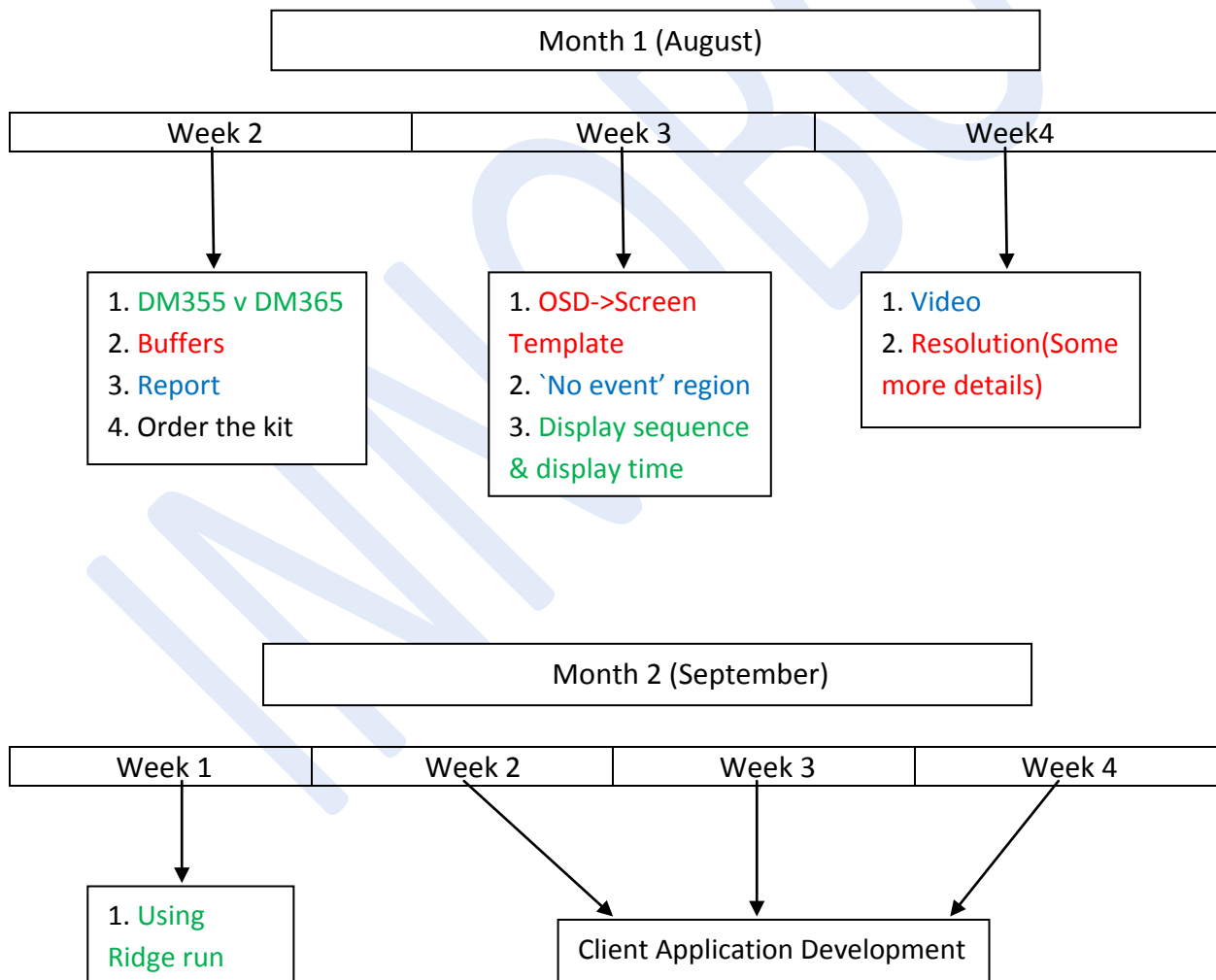
CLIENT

— Anupam Dev Goel

— Abhinav Punia

— Sahil Aggarwal

— All three



Month 3 (October)

Week 1	Week 2	Week 3	Week 4
--------	--------	--------	--------

- 1. Making custom hardware
- 2. Application Development (Auto Power off)
- 3. 3G

Month 4 (November)

Week 1	Week 2	Week 3	Week 4
--------	--------	--------	--------

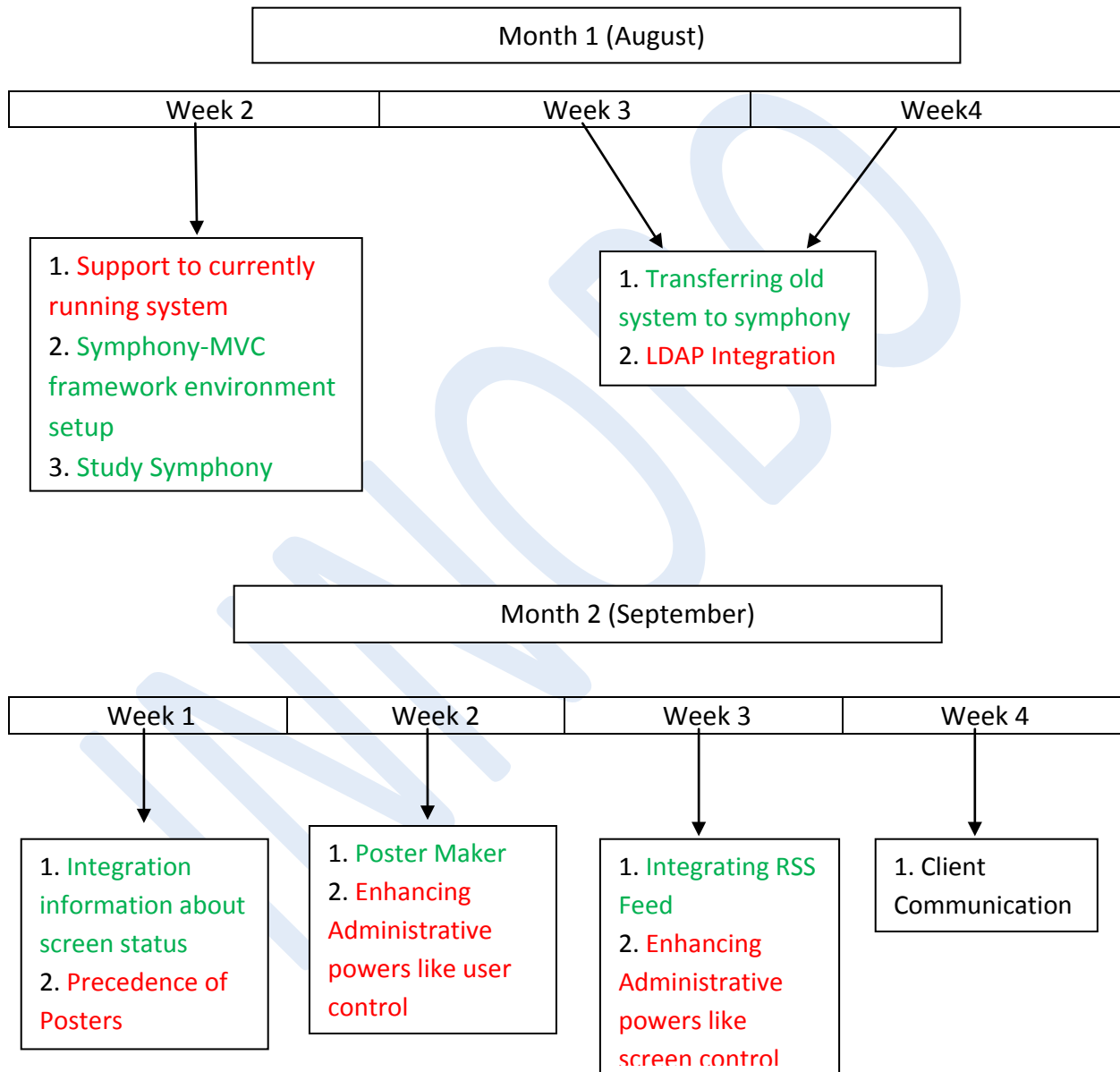
- 1. Testing new hardware
- 2. Application testing
- 3. 3G testing

SERVER

— Swapnil Jain

— Ankit Tomar

— Both



Month 3 (October)



1. Changing the view of the user interface

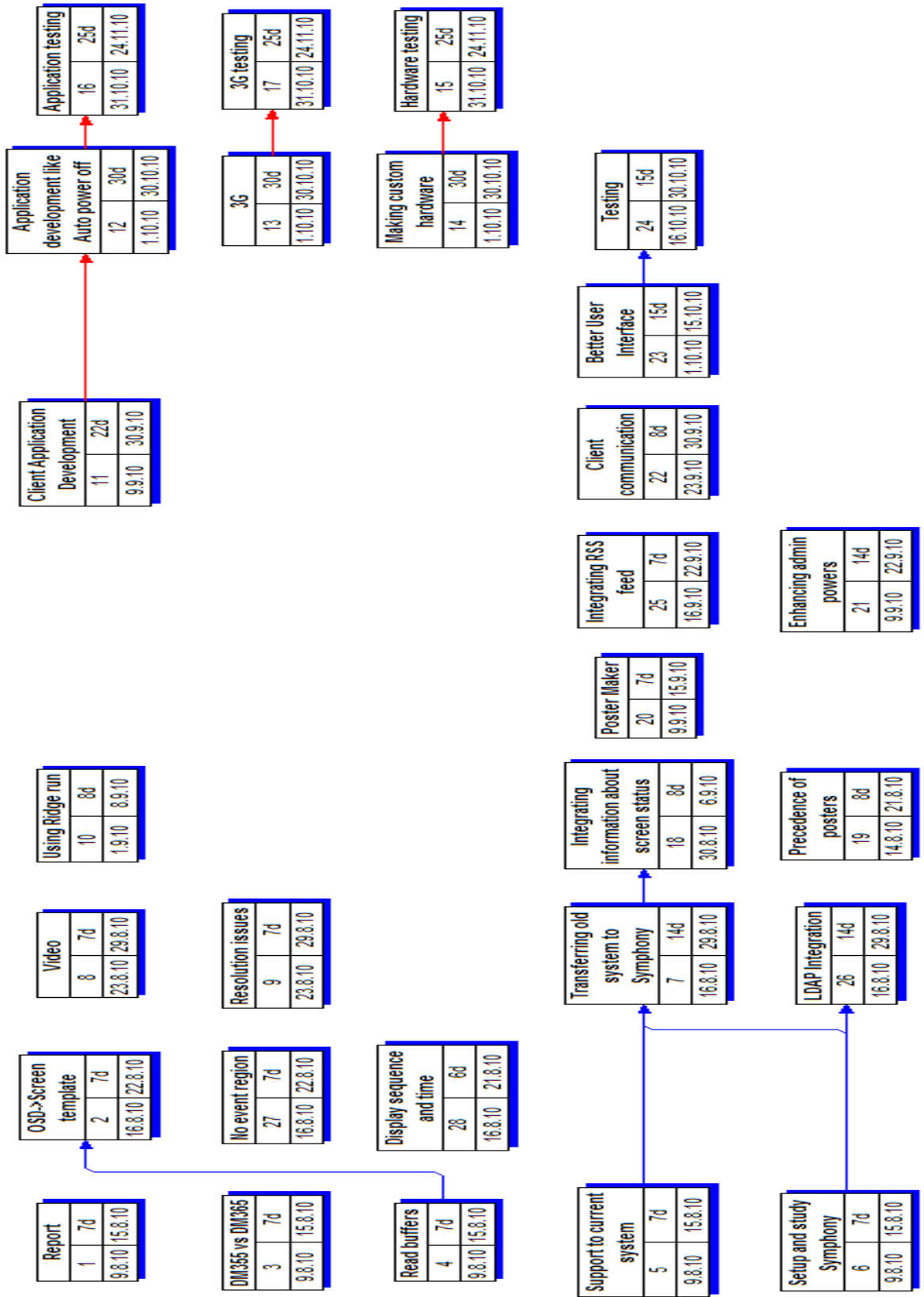
1. Testing

Month 4 (November)



Buffer Month

PERT CHART



Web link of our project

<https://sites.google.com/site/innoboextend/>

INNORO