Group Review Meetings & Final Presentation

Group Deliverable for Team C

Contents

Meeting Agendas and Minutes 3
Meeting 1 3
Agenda 3
Minutes 4
Meeting 2 7
Agenda 7
Minutes 8
Meeting 3 11
Agenda 11
Minutes 12
Meeting 4 15
Agenda 15
Minutes 16
Meeting 5 18
Agenda 18
Minutes 19
Meeting 6 23
Agenda 23
Minutes 24
Work Allocation Plan 27
Work Breakdown Structure 27
Gantt Chart 28
Functional Specification 29
Requirements 29
Use Case Diagram 29
Three separate, Functional Specifications for each subgroup including Context Diagrams, System Diagrams, Class Diagrams, ERDs, FSDs or OSDs where appropriate 30
Overall System Diagram 30
Handheld Client Device 31
Abstract Top-Level FSD 31
Keypad & Screen FSD 32
Audio Streaming Server 33

Threads Creation FSD 33
Server-Communication FSD 34
Client-Communication FSD 35
Website Administrative User Interface and Support Database 36
Website FSD 36
Database UML Diagram 39
Communications/protocol specifications 40
Messages Commands 40
Status Codes 40
Messages and URI 41
Message 41
URI 41
Message Request 41
Message Response 41
Version Control Logs 42
Makefiles 46

Meeting Agendas and Minutes

Meeting 1

Agenda

Group-C	**Meeting number**: 1
Meeting date: 24/10/2013	**Meeting start time**: 13:00
Agenda taken by: Sami Giacaman	**Meeting end time**: 14:00

Meeting Objectives:

1.	Interpretation of assignment
2.	Methodology
3.	Subgroups- Tasks and Roles
4.	Distribute tasks for the forthcoming week

Requested Attendance:

Sami Giacaman	Edward Bradley	Mark Randall
Steven Square	Ben Holden	
Nathan Toft	Sam Stone	

Discussion/Decision Items:

	Topics
1	Discuss the brief
2	Group discussion to decide on the next step forward.
3	Teams allocation –Roles of each team
4	Determine the next step in the project

Next Meeting

Meeting date:	7/11/2013
Meeting start time:	13:00
Meeting end time:	15:00
Meeting Location:	Media Editing Room 4D28

Minutes

Group-C	**Meeting number:** 1
Meeting date: 24/10/2013	**Meeting starttime:** 13:00
Minutes takenby: Sami Giacaman	**Meeting endtime:** 14:00

Meeting Objectives:

1.	Interpretation of assignment
2.	Methodology
3.	Subgroups- Tasks and Roles
4.	Distribute tasks for the forthcoming week

Members Present:

Sami Giacaman		Edward Bradley		Mark Randall	
Steven Square		Ben Holden			
Nathan Toft		Sam Stone			

Apologies:

Name	Reason	Action Taken
—	—	—
—	—	—
—	—	—

Agenda Items:

	Topics Covered
1	Each person expresses his understanding of the brief
2	Group discussion to decide on the next step forward
3	Group discussion to allocate subgroups tasks and roles
4	Allocate work amongst the group for the forthcoming week

1

Decision Points:

Agenda Item	Discussion
1	Everyone to confirm that they have read and understood the brief. Any questions on the brief can be asked now. Ask the module tutor for any assistance if required.
2	Decide as a group what exactly the brief requires, produce a project layout, and time plan. Again utilise the module tutor for assistance.
3	Split into subgroups as instructed in the brief, based on our backgrounds, three subgroups (Embedded, Database + Server, Communication + Protocols)
4	Research work needs to be undertaken in preparation for the following week.

Actions from Previous Minutes

Document number	Task	Owner	Deadline	Result
Not applicable as this is the first formal meeting of this project.				

Teams Allocation:

- **Embedded Subgroup:**
  1-Nathan Toft
  2-Steven Square
  3-Ben Holden
- **Database + Server Subgroup:**
  1-Edward Bradly
  2-Sam Stone
- **Communication + Protocols Subgroup:**
  1-Sami Giacaman
  2-Mark Randall

New Action Points

Document number	Task	Owner	Deadline
___	Project Outline	All	31/10/2013
___	Protocols research	Mark + Sami	31/10/2013
___	Drop Box Sort out	Steven/Ben	31/10/2013
___	Git repository amazon EC2 Server	Steven	31/10/2013

2

___	Git guidance document	Steven	31/10/2013
___	Database research	Ed + Sam	31/10/2013
___	Exploring the hardware	St + NT + BH	31/10/2013
___	Research G-Stream	___	___
___	Kernel Ready image	___	___
___	Individual and group Diagrams	___	___

Next meeting agreed

Meeting date:	07/11/2013
Meeting start time:	13:00
Meeting end time:	15:00
Meeting Location:	Media Editing Room 4D28

3

Meeting 2

Agenda

Group-C	**Meeting number**: 2
Meeting date: 07/11/2013	**Meeting start time**: 13:00
Agenda taken by: Sami Giacaman	**Meeting end time**: 15:00

Meeting Objectives:

1.	Decide on Protocols.
2.	Discuss Database Design.
3.	System Requirements.
4.	Distribute tasks for the forthcoming week.

Requested Attendance:

Sami Giacaman	Edward Bradley	Mark Randall
Steven Square	Ben Holden	
Nathan Toft	Sam Stone	

Discussion/Decision Items:

	Topics
1	Present protocols to be used.
2	Discuss database implementation and design.
3	Present project plan and agree on.
4	Determine the next step in the project.

Next Meeting

Meeting date:	14/11/2013
Meeting start time:	13:00
Meeting end time:	14:00
Meeting Location:	Media Editing Room 4D28

Minutes

Group-C	**Meeting number:** 2
Meeting date: 07/11/2013	**Meeting starttime:** 13:00
Minutes takenby: Sami Giacaman	**Meeting endtime:** 14:00

Meeting Objectives:

1.	Decide on Protocols.
2.	Discuss Database Design.
3.	System Requirements.
4.	Distribute tasks for the forthcoming week.

Members Present:

Sami Giacaman		Edward Bradley		Mark Randall	
Steven Square		Ben Holden			
Nathan Toft		Sam Stone			

Apologies:

Name	Reason	Action Taken
—	—	—
—	—	—
—	—	—

Agenda Items:

	Topics Covered
1	Which protocols are we going to use for communication and audio streaming
2	Database design and implementation- frontend- backend
3	Work allocation plan and functional specification
4	Allocate work amongst the group for the forthcoming week

1

Decision Points:

Agenda Item	Discussion
1	After some discussion, it was decide that we are going to use Gstreamer for audio streaming from server to the board, and soundplus as a sound proxy at both sides. For communication between boards and the server, the use of C-MYSQL API is not going to be applicable, so we decided to write our own control protocol using HTTP RFC, in C++ to support MYSQL.
2	We are going to use MYSQL for our database with a frontend backend interaction; further details will be included in the functional specification document.
3	By the 5th of December we are required to submit a work allocation plan and a functional specification, the work allocation plan was done during the meeting, and ED is going to tidy it up, meanwhile the functional specification is going to be divided between the Database + Server Subgroup.
4	Tasks were distributed between the subgroups, each based on its speciality, and further details are on New Actions Table.

Actions from Previous Minutes

Document number	Task	Owner	Deadline
___	Git repository amazon EC2 Server	Steven	Met
___	Git guidance document	Steven	___
___	Drop Box Sort out	Steven	Met
___	Kernel Image Ready	ST + NT	___
___	TCP/IP Working	Nathan	___

2

New Action Points

Document number	Task	Owner	Deadline
___	Functional Specification	Db + Sr Group	1/12/2013
___	Work Allocation Plan	Mark + Sami	1/12/2013
___	Gstreamer on Board	EM + COM Group	20/11/2013
___	Server-Client Control Protocol	Mark + Sami	25/11/2013

Next meeting agreed

Meeting date:	14/11/2013
Meeting start time:	13:00
Meeting end time:	14:15
Meeting Location:	4D38

3

Meeting 3

Agenda

Group-C	**Meeting number:** 3
Meeting date: 14/11/2013	**Meeting start time**: 13:00
Agenda taken by: Sami Giacaman	**Meeting end time**: 14:15

Meeting Objectives:

1.	Discuss teams progress.
2.	Review produced documents.
3.	Distribute tasks for the forthcoming week.

Requested Attendance:

Sami Giacaman	Edward Bradley	Mark Randall
Steven Square	Ben Holden	
Nathan Toft	Sam Stone	

Discussion/Decision Items:

	Topics
1	Discussion of progress of work by each subgroup.
2	Present and discuss produced documents.
3	Determine the next step in the project.

Next Meeting

Meeting date:	28/11/2013
Meeting start time:	13:00
Meeting end time:	14:00
Meeting Location:	--

Minutes

Group-C	**Meeting number:** 3
Meeting date: 14/11/2013	**Meeting starttime:** 13:00
Minutes takenby: Sami Giacaman	**Meeting endtime:** 14:15

Meeting Objectives:

1.	Discuss teams progress.
2.	Review produced documents
3.	Distribute tasks for the forthcoming week.

Members Present:

Sami Giacaman		Edward Bradley		Mark Randall	
Steven Square		Ben Holden	X		
Nathan Toft		Sam Stone			

Apologies:

Name	Reason	Action Taken
Ben Holden	Coursework deadline. Been up all night.	Accepted
—	—	—

Agenda Items:

	Topics Covered
1	Discussion on work progress by each subgroup.
2	Discussion about different parts of the prototype system.
3	Allocation of work amongst the group for the forthcoming week.

1

Decision Points:

Agenda Item	Discussion
1	A simple presentation by each subgroup describing their own progress regarding the work that they were allocated to. Sam presented his team's progress on the functional specification and use case, meanwhile Steven and Nathan talked about the progress been done on installing Gsteamer on the board, and many different areas that we should consider in our implementation, Mark and Sami presented the communication protocol they are working on.
2	Different Sections of the prototype system were discussed, such how to initialize the server and the board. A decision has been made on using C++ as the server client language instead of C, for the different libraries it supports and advanced parsing.
3	New tasks were distributed between the three subgroups, each based on its speciality, and further details are on New Actions Table.

Actions from Previous Minutes

Document Number	Task	Owner	Deadline
___	Git guidance document	Steven	Met
___	Drop Box Sort out	Steven	Met
___	Kernel Image Ready	ST + NT	Met
___	TCP/IP Working	Nathan	Met
___	Messages and status codes for Server-Client	Mark	Met

2

New Action Points

Document Number	Task	Owner	Deadline
___	Functional Specification	Db Group	1/12/2013
___	Work Allocation Plan	Edward	1/12/2013
___	Gstreamer on Board	EM Group	20/11/2013
___	Server-Client Control Protocol	Mark + Sami	25/11/2013
___	Multiple FSDs of our system	Mark + Sami	1/12/2013
___	References Document	Sami	1/12/2013

Next meeting agreed

Meeting date:	28/11/2013
Meeting start time:	13:00
Meeting end time:	14:15
Meeting Location:	--

3

Meeting 4

Agenda

Group-C	**Meeting number:** 4
Meeting date: 28/11/2013	**Meeting start time**: 13:00
Agenda taken by: Sami Giacaman	**Meeting end time**: 14:15

Meeting Objectives:

1.	Evaluation of documents produced by each subgroup.
2.	Final discussion and modification on the system and documents.
3.	Distribute tasks for the forthcoming week.

Requested Attendance:

Sami Giacaman	Edward Bradley	Mark Randall
Steven Square	Ben Holden	
Nathan Toft	Sam Stone	

Discussion/Decision Items:

	Topics
1	An evaluation of documents to be submitted.
2	Final discussion about the prototype system.
3	Determine the next step in the project.

Next Meeting

Meeting date:	05/12/2013
Meeting start time:	13:00
Meeting end time:	14:00
Meeting Location:	--

Minutes

Group-C	**Meeting number:** 4
Meeting date: 28/11/2013	**Meeting starttime:**13:00
Minutes takenby: Mark Randall	**Meeting endtime:**14:00

Meeting Objectives:

1.	Evaluation of documents to be submitted
2.	Final discussion about the prototype system
3.	Determine the next step in the project

Members Present:

Sami Giacaman	N	Edward Bradley	Y	Mark Randall	Y
Steven Square	Y	Ben Holden	Y		
Nathan Toft	Y	Sam Stone	N		

Apologies:

Name	Reason	Action Taken
Sami	Food Poisoning	Accepted
Sam	Coursework Deadlines overlapped	Accepted
—	—	—

Agenda Items:

	Topics Covered
1	Hand in has been discussed, and a details are available of what to submit on dropbox
2	Demostration on 12th December, and what to have ready for this
3	Git issues

Decision Points:

Agenda Item	Discussion
1	Discussed what to submit for the 5th December document hand in. Everyone knows what we are submitting, and a document has been put into dropbox detailing this (written by Ed)
2	Server and Client is ready for the prototype, and the Igep has audio playback working. We now need to add gstream initialisation onto client and server, as well as getting the client onto the Igep.
3	We also discussed potentially adding extra features for the prototype (like authenticaiton), however these shall be considered a stretch goal and only attempted when we have got the client and server initialising gstreamer
4	Gstreamer has proved to be more time consuming than originally thought. Steve, Nathan and Ben will need assitance from Mark and Sami. Firstly to know where to put the gstreamer commands, and secondly to create threads for streaming.

Actions from Previous Minutes

Document number	Task	Owner	Deadline	Result
___	Gstreamer on board	Em Group	20/11/2013	Not Complete
___	Server-Client Protocol	Mark	25/11/2013	Completed
___	Server-Client programs	Mark+Sami	28/11/2013	Completed

New Action Points

Document number	Task	Owner	Deadline
___	References for December Hand in	Ed	5/12/2013
___	Gstreamer integrated into Server-Client	Em Team	11/12/2013
___	Fix Git issues...	Steve	At his discression

Next meeting has not yet been decided

There will be no meeting on 5th December due to Coursework Hand Ins

Meeting 5

Agenda

Group-C	**Meeting number:** 5
Meeting date: 30/01/2014	**Meeting start time**: 11:00
Agenda taken by: Sami Giacaman	**Meeting end time**: 12:00

Meeting Objectives:

1.	Review teams progress.
2.	Refine project plan estimates.
3.	Distribute tasks for the forthcoming week.

Requested Attendance:

Sami Giacaman	Edward Bradley	Mark Randall
Steven Square	Ben Holden	
Nathan Toft	Sam Stone	

Discussion/Decision Items:

	Topics
1	Discussion on multicasting.
2	Discussion about database design.
3	Discussion about web interface.

Next Meeting

Meeting date:	06/02/2014
Meeting start time:	10:00
Meeting end time:	11:00
Meeting Location:	--

Minutes

Group-C	**Meeting number:** 5
Meeting date: 30/01/2014	**Meeting starttime**: 11:00
Minutes takenby: Edward Bradley	**Meeting endtime**: 12:00

Meeting Objectives:

1.	Review teams progress.
2.	Refine project plan estimates
3.	Distribute tasks for the forthcoming week.

Members Present:

Sami Giacaman	✓	Edward Bradley	✓	Mark Randall	✓
Steven Square		Ben Holden			
Nathan Toft	✓	Sam Stone			

Apologies:

Name	Reason	Action Taken
Sam Stone	Family Problems	Accepted
—	—	—

Agenda Items:

Topics Covered	
1	Discussion on multicasting.
2	Discussion about database design
3	Discussion about web interface .

1

Decision Points:

Agenda Item	Discussion
1	Mark and Steve have already discussed lots of the multicasting implementation details separately. One remaining issue was that when in a multicast group, the 'slave' devices should not be able to play/pause – only the 'master' device can play/pause (this is so that the rebel child in a class does not pause the track at the start and not ever listen to it etc.) This issue had been troubling Mark and Steve, however Edward came up with a simple solution: When a IGEP (handset) enters the 'slave' state (by having a multicast activation code typed in), its keypad is disabled. Thus the user could press play/pause as much as they want, but nothing will happen. To exit the multicast 'slave' mode the original intention was to have a special secret code typed in (by the receptionist), but the idea now is to perform this over the USB or RS-232 port. The same approach will be used for a regular (non multicast) handset – to reset the device ready for the next user, the receptionist will send a command over the to perform this over the USB or RS-232 port. This is mapped to the real-world scenario where an audio guide in a museum is returned to a charging rack/bay, which would connect to the USB or RS-232 port to send a reset command as-well as connecting the battery terminals onto charge.

2

2	Some initial database design had been completed before the meeting, and additions were made during the meeting. Particular items were for a means to identify each IGEP handset (MAC Address or serial number), and to handle multiple versions of each exhibit track (based on which language and expertise level the customer was registered as). The database design derived in the meeting is not finalised and is still open for discussion. **Table1 (IGEPs):** Mac Address Authorisation Code Use Status IP Address of IGEP Handset Customer ID **Table2: (Customers)** Customer ID Customer Name Customer Address Expertise Level Chosen Language Chosen MAC Address of IGEP Handset they used **Table3 (Exhibits):** Exhibit ID Exhibit Location Exhibit Name Audio Code **Table4 (Audio Files):** Audio Code Language Difficulty File path on server **Table 5 (statistical logging):** Customer ID Exhibit ID DateTime [Note that this needs to be turned into an ER Diagram]
3	Sam had given apologies for not being able to attend this meeting, and also had not attended the previous meeting. Edward will speak with Sam about the progress web interface he was creating. Also a reminder that code should be on Git if possible.

3

Actions from Previous Minutes

Document Number	Task	Owner	Deadline
___	Reference for December Hand In	Edward	Met
___	Gstreamer integrated into Server-Client	All embedded	Met
___	Fix Git issues	Steve	Unknown

New Action Points

Document Number	Task	Owner	Deadline
___	Speak with Sam regarding web interface and database.	Edward	06/02/2014
___	Speak with Ben regarding progress	Edward	06/02/2014
___	Speak with Steven regarding multicasting decision made in his absence.	Mark	06/02/2014

Next meeting agreed

Meeting date:	06/02/2014
Meeting start time:	10:00
Meeting end time:	11:00
Meeting Location:	Lab Tutorial

4

Meeting 6

Agenda

Group-C	**Meeting number:** 6
Meeting date: 24/04/2014	**Meeting start time:** 13:00
Agenda taken by: Ben Holden	**Meeting end time:** 14:00

Meeting Objectives:

1	Discuss teams progress.
2	Confirm members understand requirements for group hand in.
3	Discuss what is required for demonstration.

Requested Attendance:

Sami Giacaman	Edward Bradley	Mark Randall
Steven Square	Ben Holden	
Nathan Toft	Sam Stone	

Discussion/Decision Items:

	Topics
1	Discussion of progress of work by each subgroup.
2	Discussion of work needed for group hand in.
3	Discussion of work needed for demonstration.

Minutes

Group-C	**Meeting number:** 6
Meeting date: 24/04/2014	**Meeting starttime:** 13:00
Minutes takenby: Ben Holden	**Meeting endtime:** 14:00

Meeting Objectives:

1.	Discuss teams progress.
2.	Confirm members understand required objectives for group hand in.
3.	Discuss what is required for demonstration.

Members Present:

Sami Giacaman	Y	Edward Bradley	N	Mark Randall	Y
Steven Square	Y	Ben Holden	Y		
Nathan Toft	Y	Sam Stone	Y		

Apologies:

Name	Reason	Action Taken
Edward Bradley	Confirmed earlier that would not be attending due to exams.	Accepted
—	—	—

Agenda Items:

Topics Covered	
1	Discussion on work progress by each subgroup.
2	Discussion of work needed for group hand in.
3	Discussion of work needed for demonstration.

1

Decision Points:

Agenda Item	Discussion
1	The progress of each group was discussed. Due to other coursework commitments, progress has been slow but the majority of the work is done.
2	The work needed for the group hand in was also discussed. The embedded group need to produce FSDs and user interaction diagrams for all parts of the embedded system. The database group need to produce FSDs and UML diagrams for the database system. This should be done by the 28th of April. It was also discussed to produce a make file to compile all parts of the system. Ben has agreed to draw a system context diagram of the system. Edward has agreed via the Facebook group to compile the group report for the hand in. Each member is aware of the individual requirements for the hand in. It is agreed to meet after the exam on the 30th of April to hand in the group portion of the project.
3	The demonstration is on the 19th of May so the system should be ready to show by then. This requires all parts of the system to be working.

Actions from Previous Minutes

Document Number	Task	Owner	Deadline
___	Speak with Sam regarding web interface.	Edward	Met
___	Speak with Ben regarding process	Edward	Met
___	Speak with Steven regarding multicasting decision made in his absence.	Mark	Met

2

New Action Points

Document Number	Task	Owner	Deadline
___	Ensure FSDs and user interaction diagrams for all parts of embedded system are complete.	EM Group	28/04/2014
___	Ensure FSDs and UML diagrams for all parts of Database are complete.	DB Group	28/04/2014
___	Compile Group report	Edward	30/04/2014
___	Produce Make Files for software parts of system	EM + DB Groups	30/04/2014
___	Everyone should understand and complete the requirements for the individual hand in.	Everyone	28/04/2014
___	It has been decided to hand in the group report on the 30th of April.	Everyone	30/04/2014
___	Get the system ready for demonstration	EM + DB Groups	19/05/2014

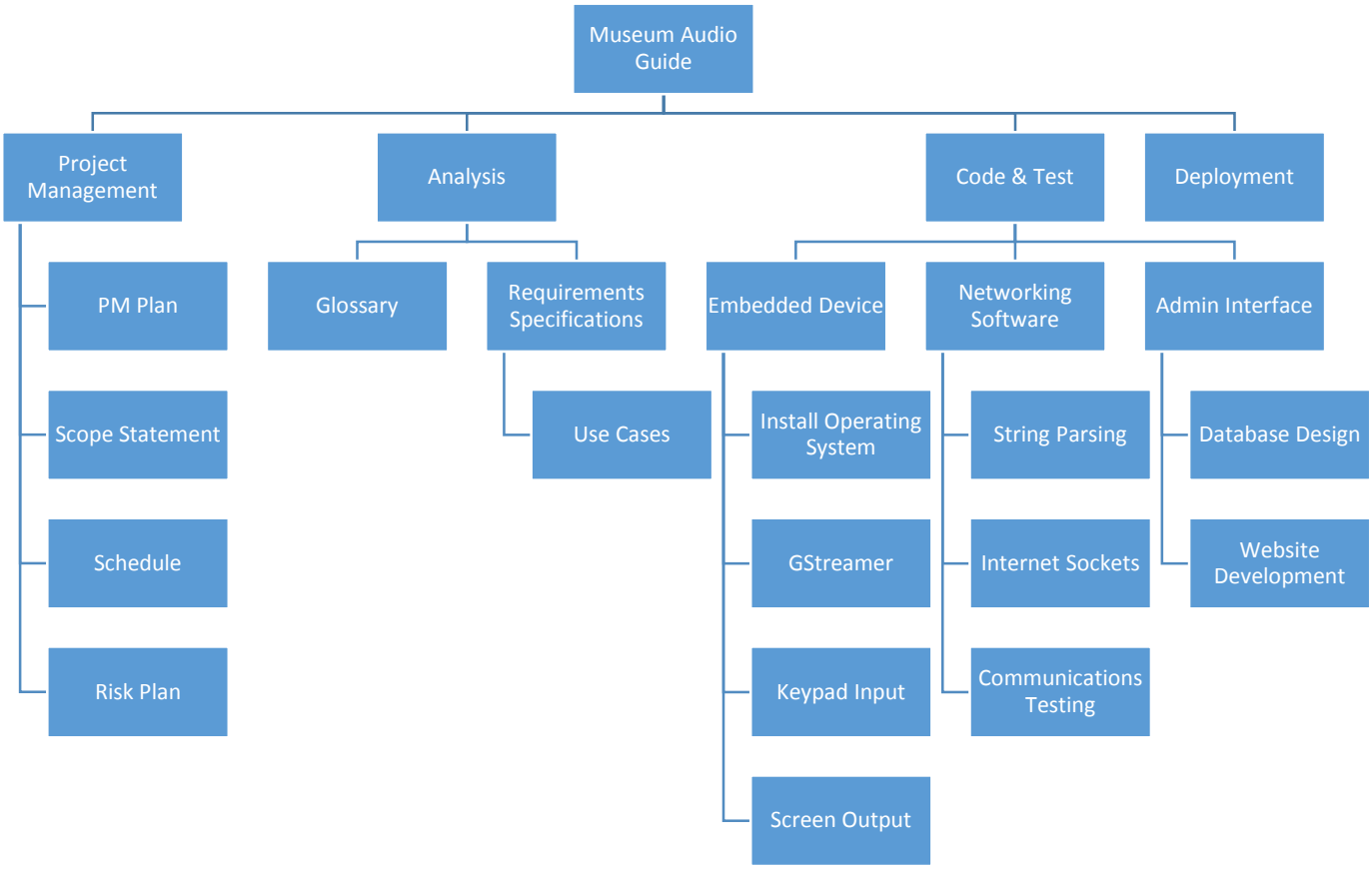
Next meeting agreed

Meeting date:	30/04/2014
Meeting start time:	12:30
Meeting end time:	13:00
Meeting Location:	--

3

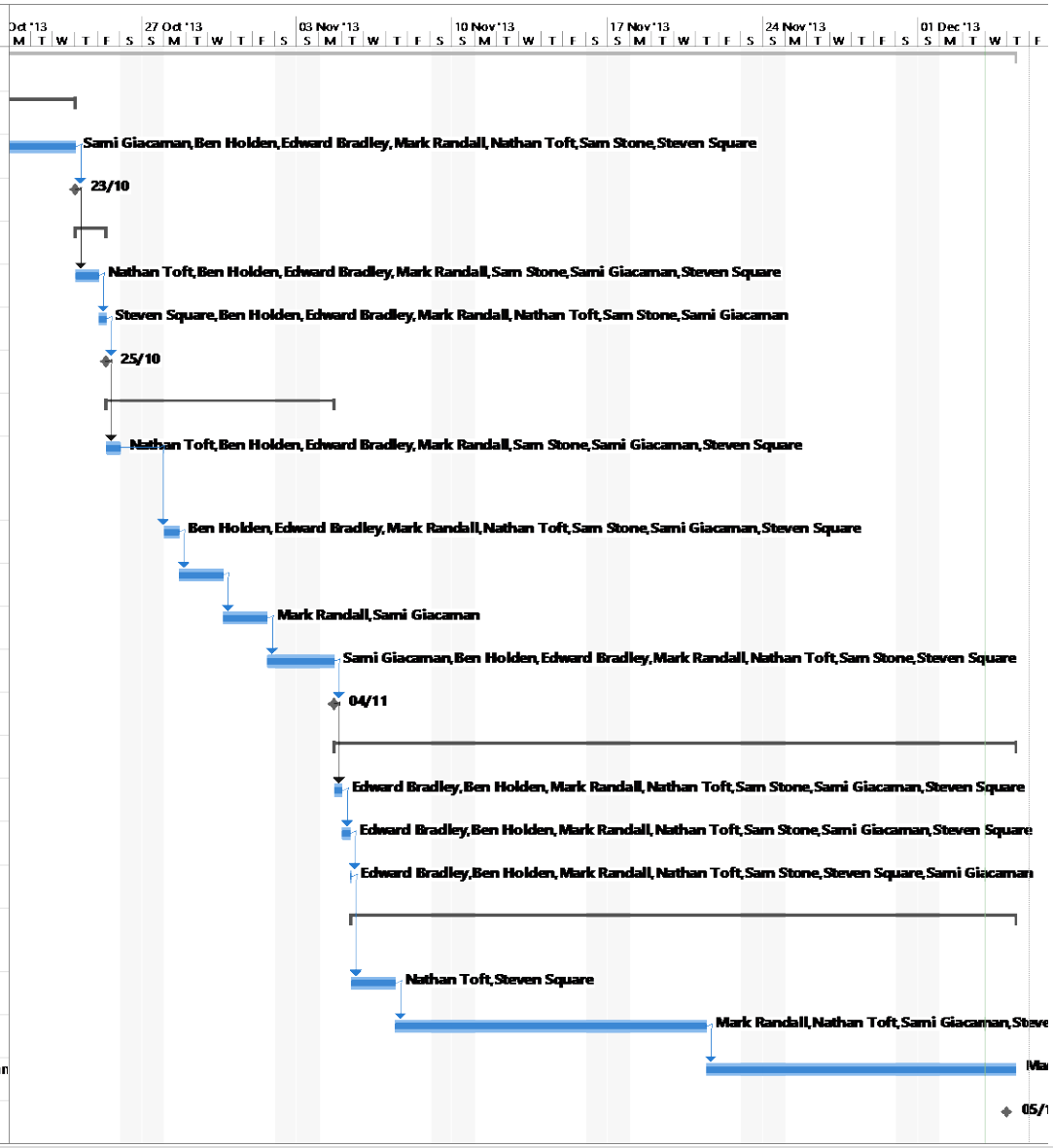
Work Allocation Plan

Work Breakdown Structure



Gantt Chart

ID	Task Mod	Task Name	Duration	Start	Finish	Predecesso	Resource Names
0		**Project Plan**	**33.25 days**	**Mon 21/10/13**	**Thu 05/12/13**		
1		**Scope**	**3 days**	**Mon 21/10/**	**Wed 23/10/1**		
2		Determine project scope	3 days	Mon 21/10/13	Wed 23/10/13		Sami Giacaman,Ben Holden,Edward
3		Scope complete	0 days	Wed 23/10/13	Wed 23/10/13	2	
4		**Analysis/Software Requirements**	**1.13 days**	**Thu 24/10/13**	**Fri 25/10/13**		
5		Define software requirements	1 day	Thu 24/10/13	Thu 24/10/13	3	Nathan Toft,Ben Holden,Edward
6		Develop delivery timeline	1 hr	Fri 25/10/13	Fri 25/10/13	5	Steven Square,Ben Holden,Edward
7		Analysis complete	0 days	Fri 25/10/13	Fri 25/10/13	6	
8		**Design**	**6.75 days**	**Fri 25/10/13**	**Mon 04/11/1**		
9		Define software specifications for embedded system	7 hrs	Fri 25/10/13	Fri 25/10/13	7	Nathan Toft,Ben Holden,Edward Bradley,Mark Randall,Sam Stone,Sami
10		Define software specifications for server	7 hrs	Mon 28/10/13	Mon 28/10/13	9	Ben Holden,Edward Bradley,Mark
11		Design Database Structure	2 days	Mon 28/10/	Wed 30/10/13	10	
12		Define Streaming & Communication Protocols	2 days	Wed 30/10/13	Fri 01/11/13	11	Mark Randall,Sami Giacaman
13		Review functional specifications	1 day	Fri 01/11/13	Mon 04/11/13	12	Sami Giacaman,Ben Holden,Edward
14		Design complete	0 days	Mon 04/11/	Mon 04/11/13	13	
15		**Development**	**22.38 days**	**Mon 04/11/**	**Thu 05/12/13**		
16		Review functional specifications	1 hr	Mon 04/11/13	Mon 04/11/13	14	Edward Bradley,Ben Holden,Mark
17		Identify modular/tiered design parameters	1 hr	Tue 05/11/13	Tue 05/11/13	16	Edward Bradley,Ben Holden,Mark
18		Assign development staff	1 hr	Tue 05/11/13	Tue 05/11/13	17	Edward Bradley,Ben Holden,Mark
19		**Develop prototype based on functional specifications**	**22 days**	**Tue 05/11/13**	**Thu 05/12/13**		
20		Install Linux on embedded device	2 days	Tue 05/11/13	Thu 07/11/13	18	Nathan Toft,Steven Square
21		MP3 encoder decoder for gstreamer	2 wks	Thu 07/11/13	Thu 21/11/13	20	Mark Randall,Nathan Toft,Sami
22		Develop basic server	2 wks	Thu 21/11/1	Thu 05/12/13	21	Mark Randall,Sami Giacan
23		Prototype ready for demonstration	0 days	Thu 05/12/13	Thu 05/12/13		

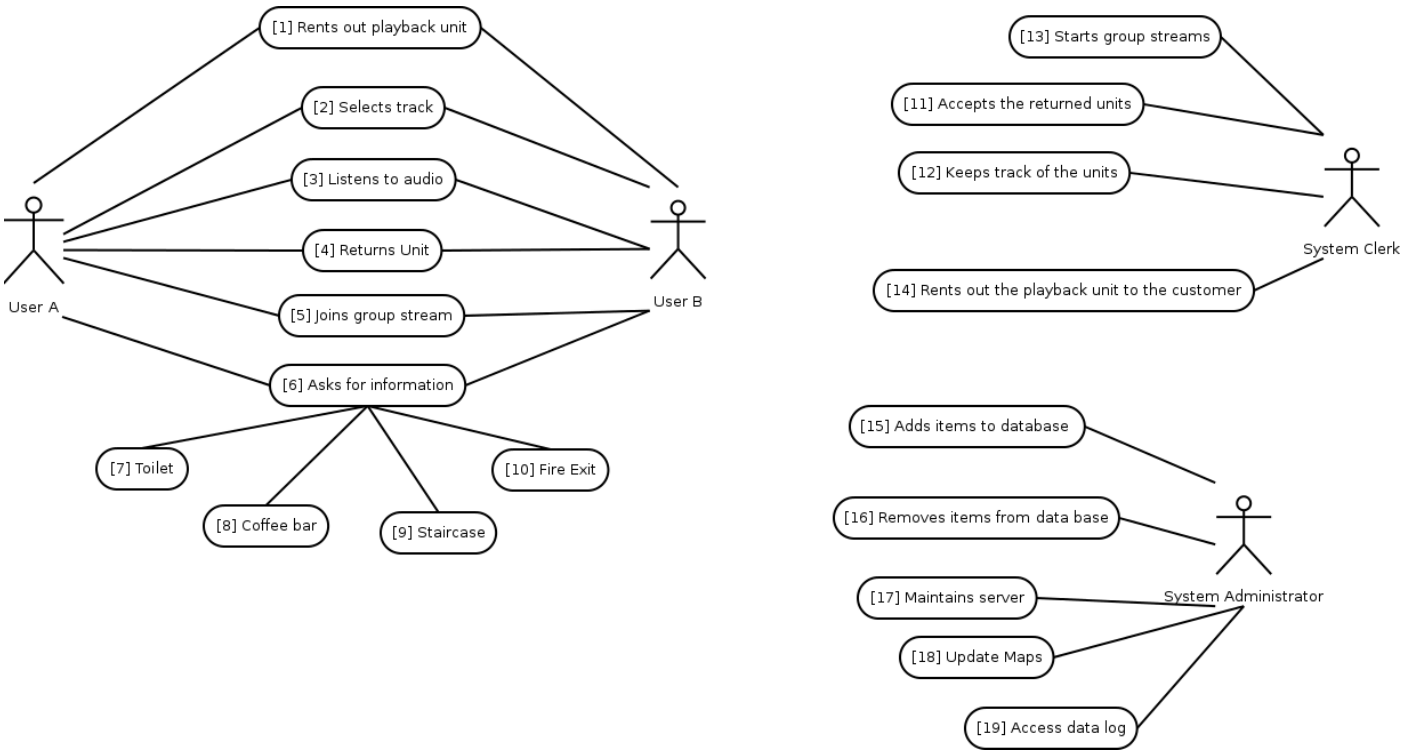


Functional Specification

Requirements

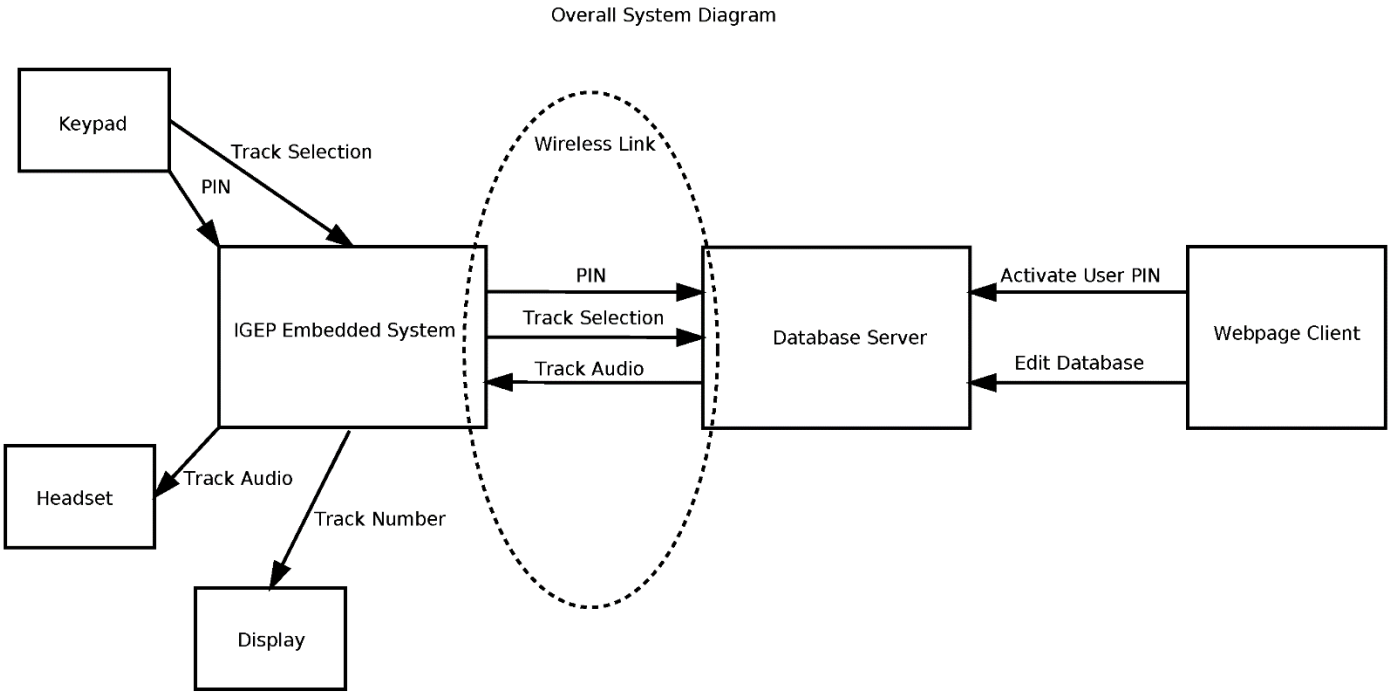
- The system must allow for the renting out of multiple handsets capable of selecting and playing audio streams.
- The system must be able to stream MP3 audio files over a network to multiple handsets simultaneously. For individual streams, play, pause fast forward and rewind functionality should be implemented.
- The handset must be able to select the audio stream based on a number or code found on the relevant exhibit.
- The system must also allow for the setup and listening of group streams, where there are multiple devices connected to the same single stream.
- The system must allow for the handsets to be kept track of whether they are rented out or in storage.
- The system should be able to store information on user if the user allows for the information to be taken, in accordance of the data protection act, 1998.
- The system should use the user's information to stream different information based on the user's background. E.g simpler streams for younger users and more advanced for older users.
- The system must be able to be updated by a system administrator, for the purpose of adding/removing audio files and updating maps and exhibits.
- The system should be able to give information to the user based on their location. Examples of this include the nearest toilet, coffee bar, staircase and fire exit.

Use Case Diagram



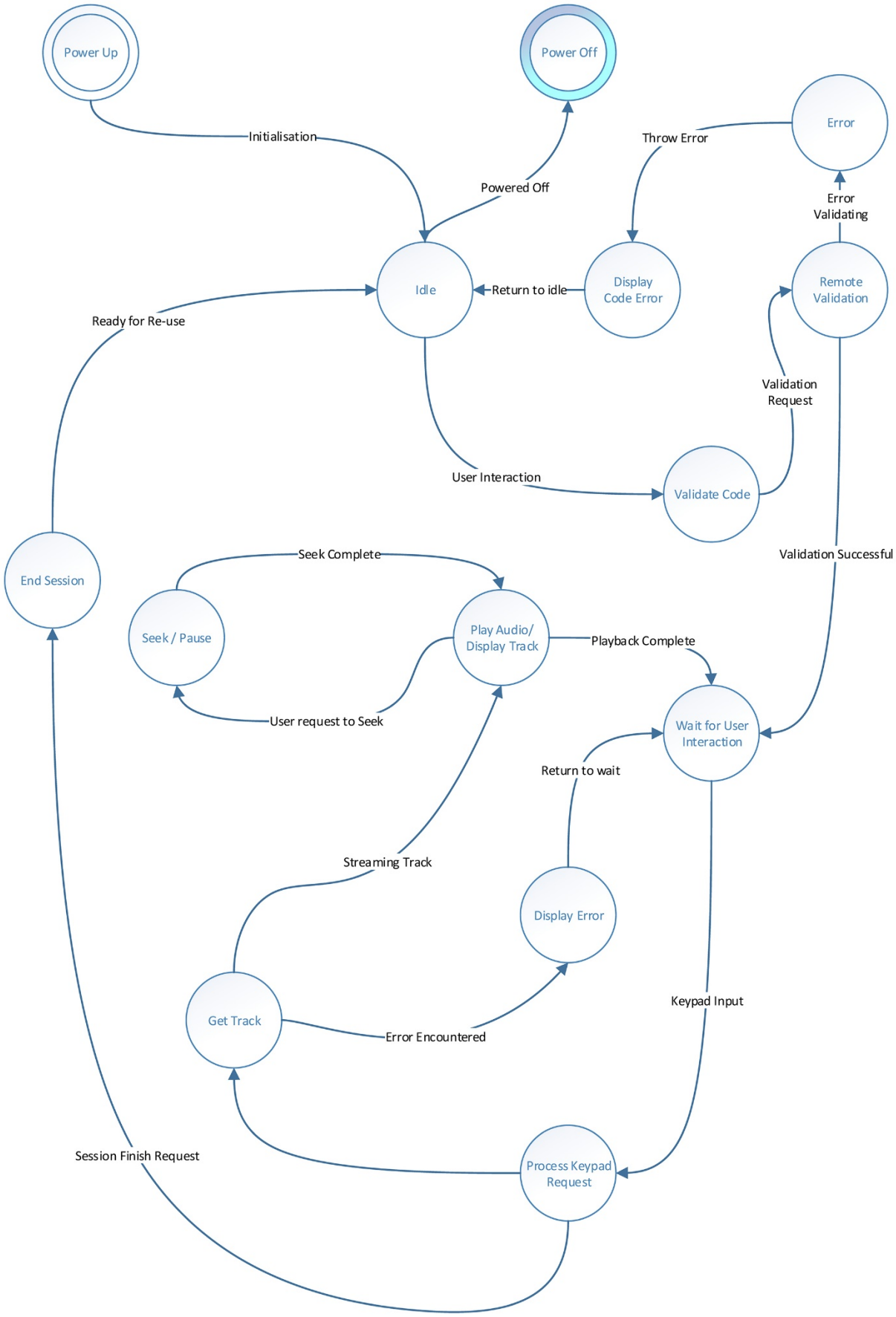
Three separate, Functional Specifications for each subgroup including Context Diagrams, System Diagrams, Class Diagrams, ERDs, FSDs or OSDs where appropriate.

Overall System Diagram

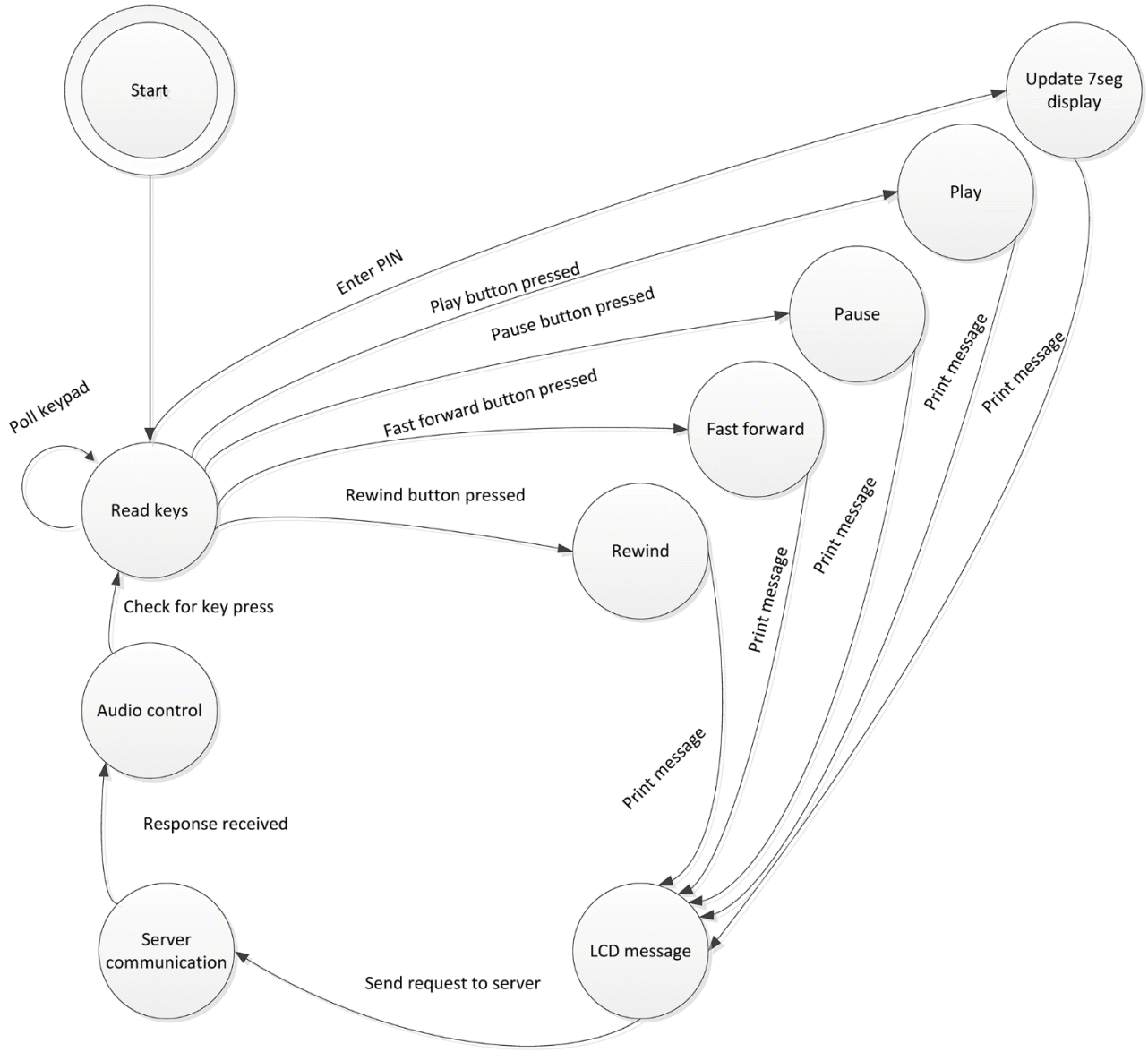


Handheld Client Device

Abstract Top-Level FSD

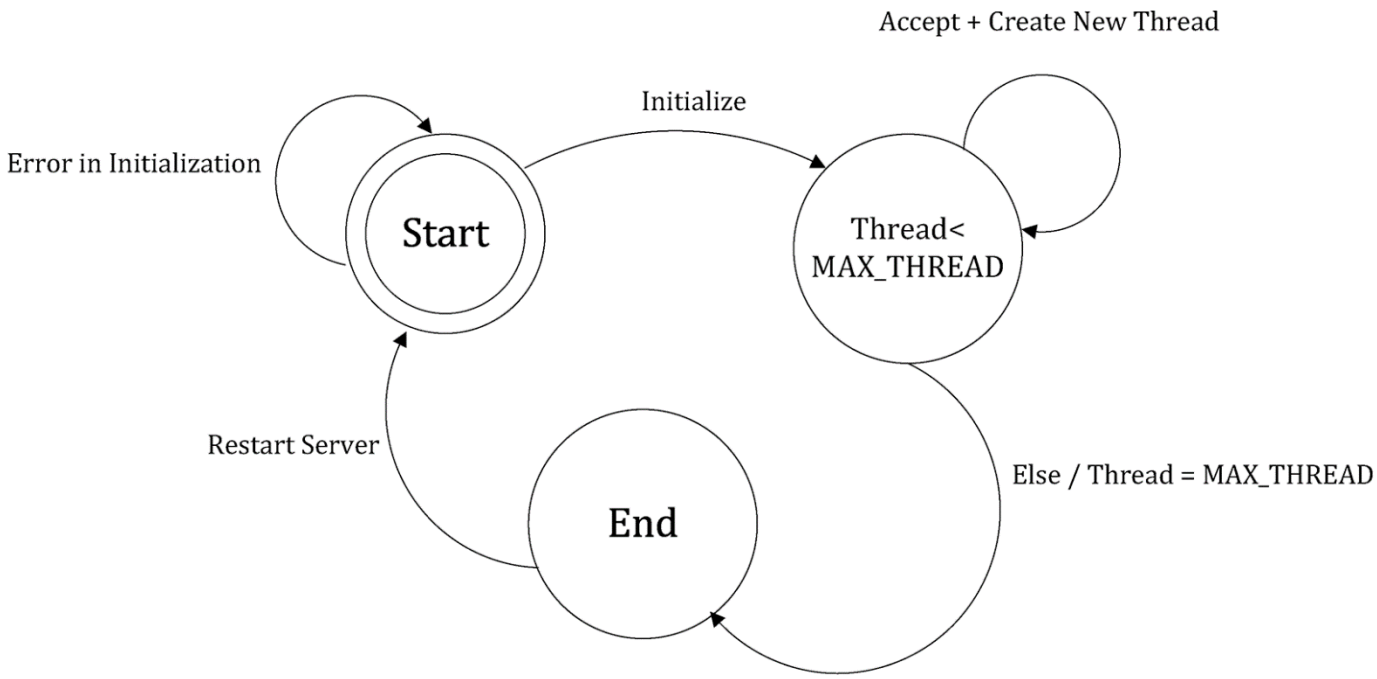


Keypad & Screen FSD

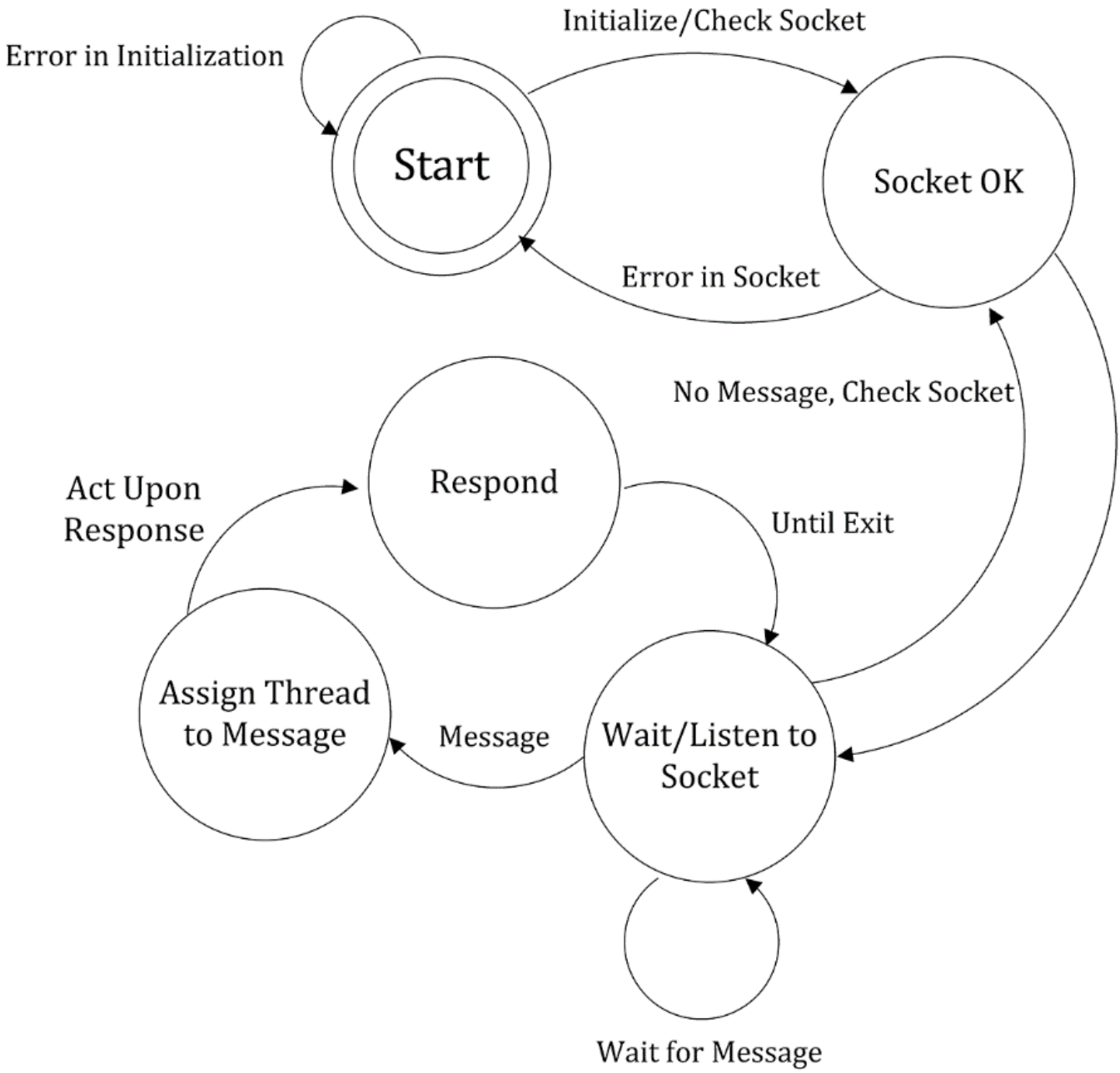


Audio Streaming Server

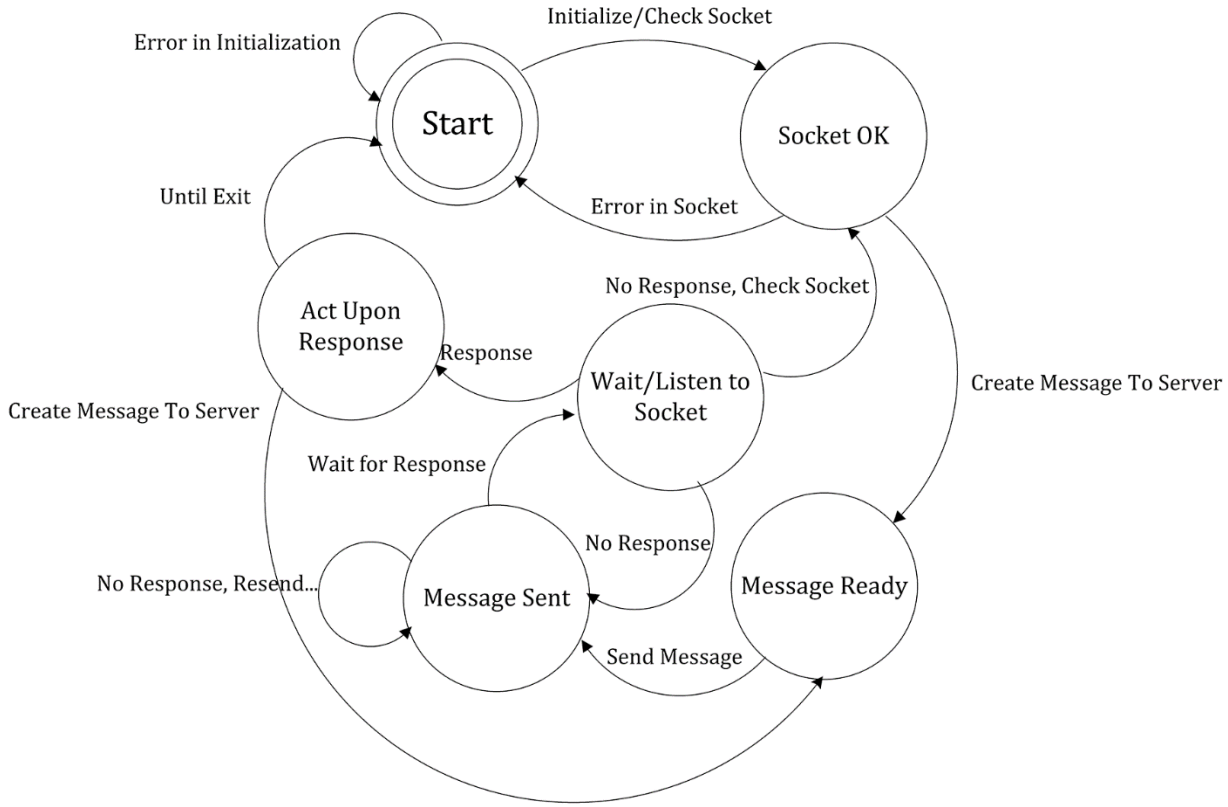
Threads Creation FSD



Server-Communication FSD



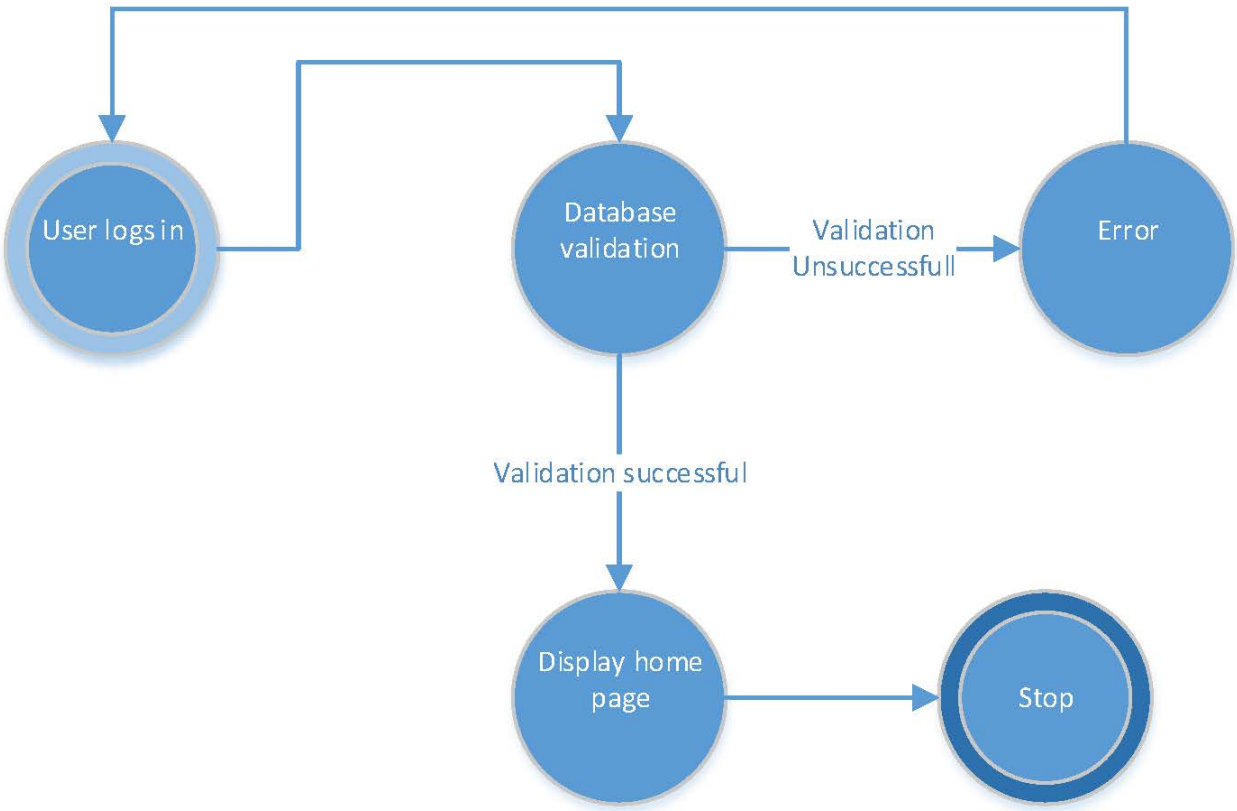
Client-Communication FSD



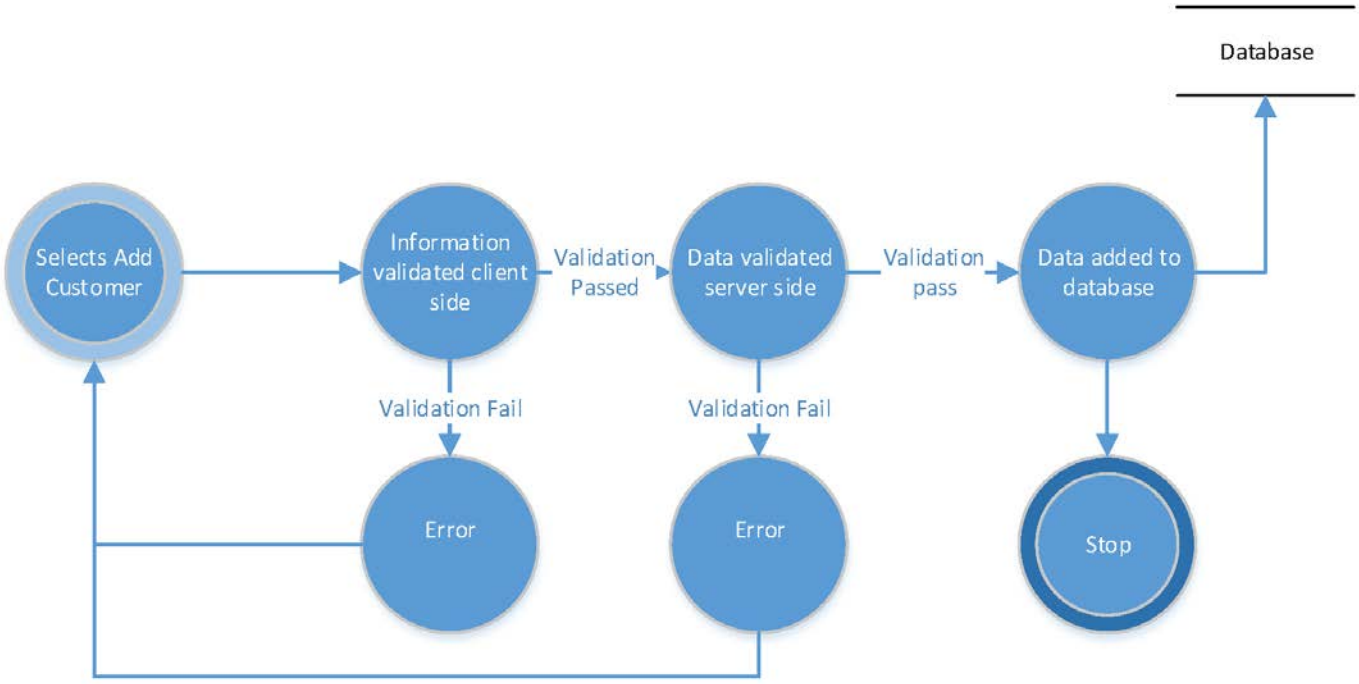
Website Administrative User Interface and Support Database

Website FSD

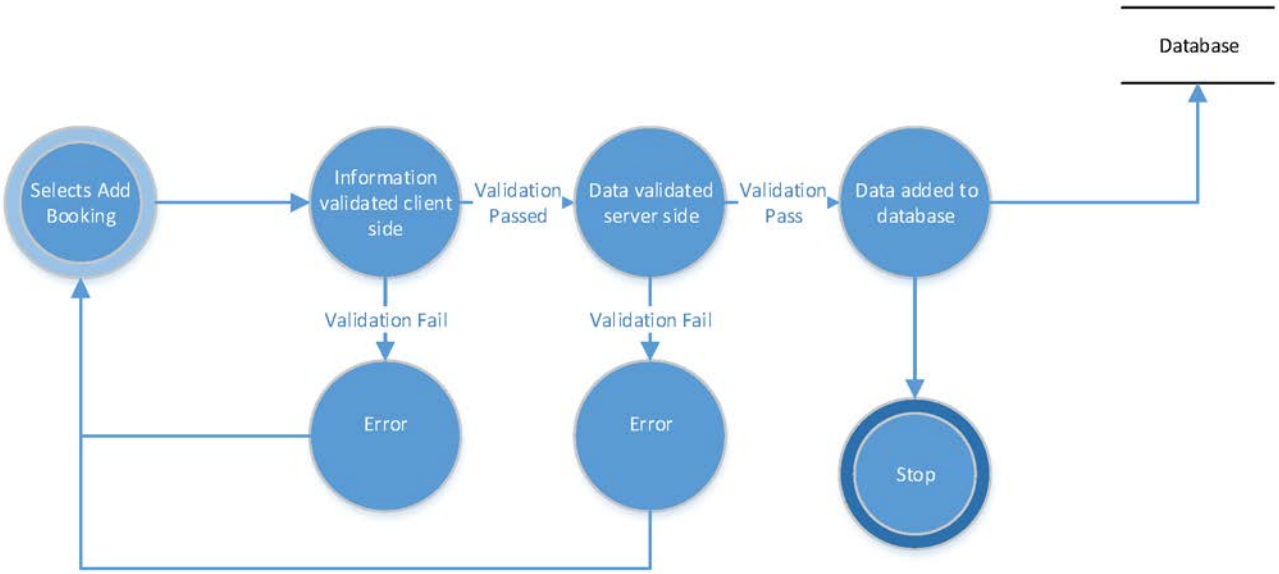
Login



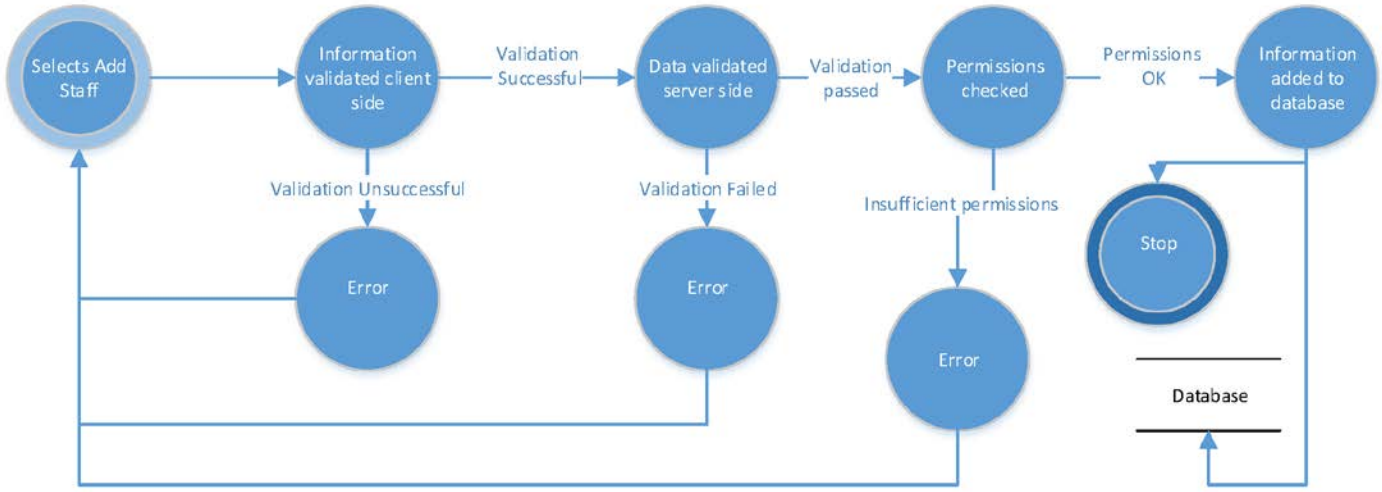
Add Customer



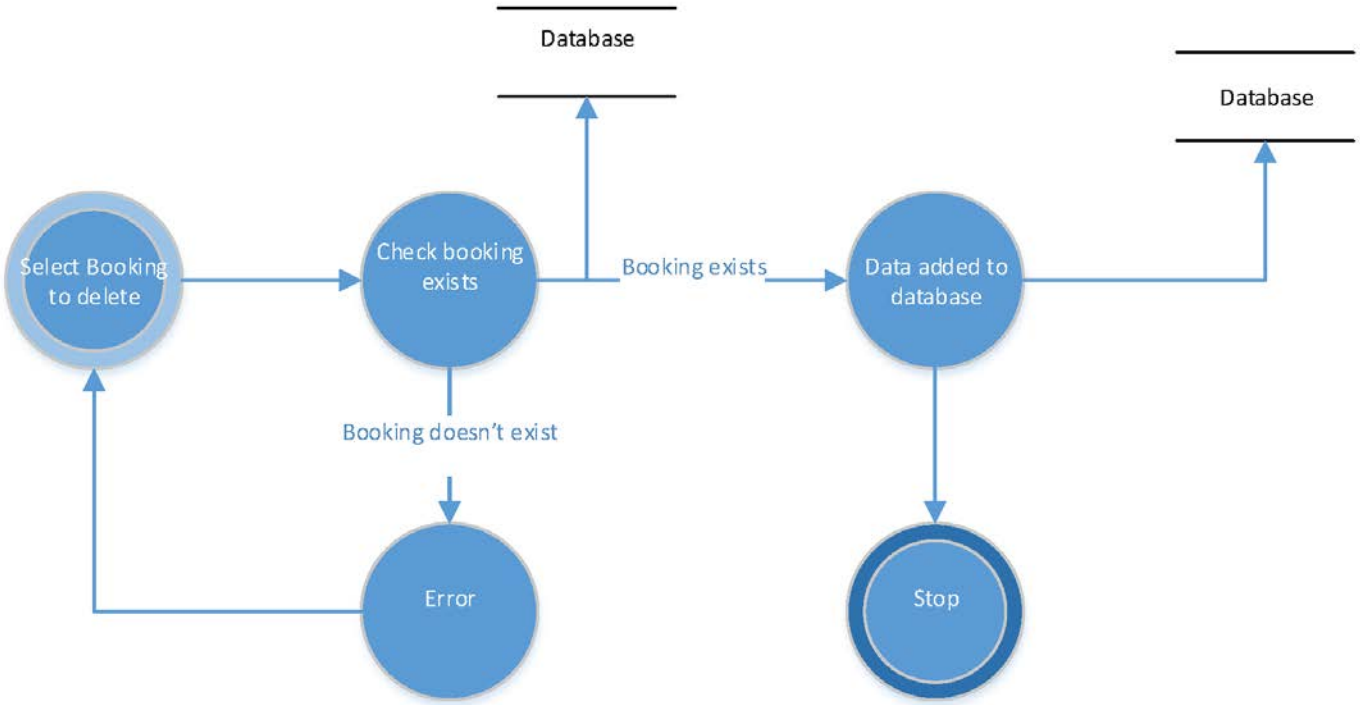
Add Booking



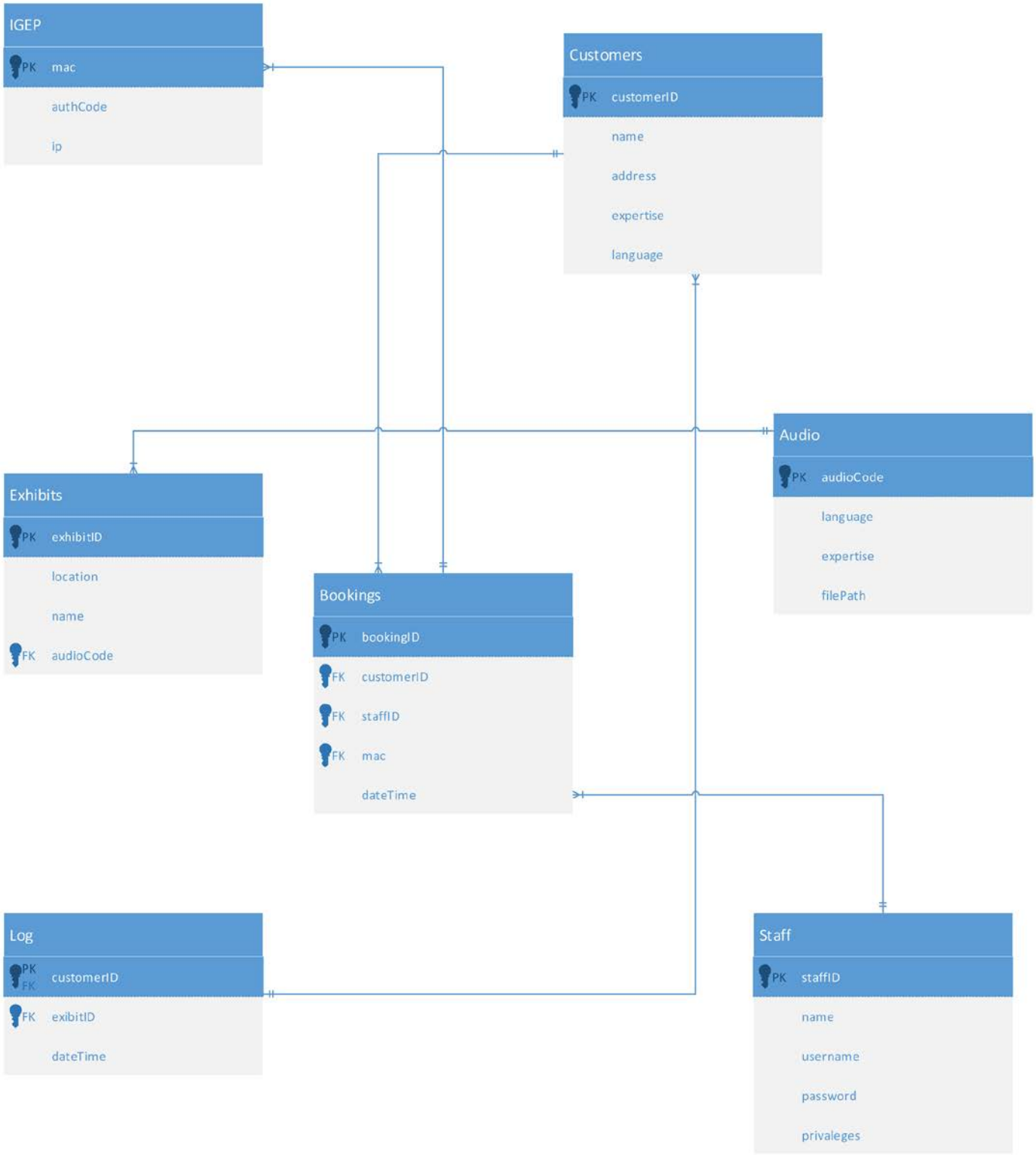
Add Staff



Delete Booking



Database UML Diagram



Communications/protocol specifications

Messages Commands

- **Connect –** Connect the Igep to the server.
- **Post –** Send purchased/activation code to server, server checks this against SQL DB, before sending an acknowledged/activation message back to the client. A flag needs to be ticked in the SQL DB to state that this code is being used.
- **Get –** Send playback code to server, server then checks this code against the SQL DB to find the address/location of the file we wish to stream. Server sends an acknowledged request message, the client (on receiving this message) enters a logic state to start the stream.
- **Delete –** The user wishes to finish using the guide. The code they originally entered is sent to the server, which then deletes that code from the SQL DB, and sends a deactivation message to the client.

Status Codes

- **200 OK -**
  - **Post -** Purchased/Activation code was accepted. On receiving this status code, the Igep activates guide functionality.
  - **Get (normal) –** Playback code was accepted, and the address/location of the file we wish to stream is sent via this message.
  - **Get (multicast) –** Playback code was accepted, and all users are available, and ready, to multicast. Address/location of the file that we wish to stream for multicast is sent via the message.
  - **Delete –** Request to end session has been accepted, the code is put into *"Limbo"* for a certain amount of time, before being made available once again for the payment system to register as available to a user. On receiving this message, the client ends the session.
- **202 Accepted –** A Get message has been sent (with an extra flag to enter Multicast mode). Until all users are ready to start multicasting, the Igep must wait for an **OK** message from the server (with the address/location contained within it for the file) before the streaming can begin.
- **400 Bad Request –** A Get message was sent, with an invalid/incorrect playback code, to the server. The Igep should display a message to the user, stating that the audio file does not exist for that code.
- **402 Payment Required –** A Post message has been sent with a purchased/activation code that has not been registered/no payment has been made. An appropriate message to the user should state that they need to pay to use the service.
- **403 Forbidden –** A Post message has been sent with a purchased/activation code, that is currently being used by another user. An appropriate message to the user should state that the current code is not available.
- **500 Internal Server Error –** This should be used if the SQL DB is not responding/not available, or if the audio files are unavailable to stream. The user should be informed appropriately. (i.e. If authentication can't be verified, tell the user that a server issue occurred, and they need to contact an assistant. If getting a audio file failed, tell the user the service is not available, and to try again, or contact an assistant. If we fail to use the Delete command, then just tell the user the session is ended, and we can send the error message to the web admin side)
- **503 Service Unavailable** – This should be used when sending a Get (multicast) message. If cannot start a multicast session (not due to the file not existing/unavailable) this should be sent.
- multicast session (not due to the file not existing/unavailable) this should be sent.

Messages and URI

Message

[*command*] [*protocol*] #[*code*]

URI

Message Request

[command] [protocol]\r\n\r\n[mac]\r\n[code]

Example:

Host: 8F:24:59:A3:74

POST HTTP/1.1

#4242

Message Response

[*protocol*] [*code*] [*title*]\r\n\r\n[*message-body*]

Example:

HTTP/1.1 200 OK

Version Control Logs

The project has made use of a distributed revision control and source code management system. The versioning control system used was Git, hosted on a private Assembla Workspace. An extract of the Git commit history is given below in reverse chronological order:

```
commit 27bb214cacd96f81c99a40c0202c43f356348c03
Author: nath247 <nathan957@hotmail.com>
Date:   Wed Apr 30 18:08:46 2014 +0100

    Few more comments

commit 4927e1097a3bd82b4432d4917a5883ad82a5a147
Merge: 2cb62d6 9972568
Author: Steve Square <steven.square@btinternet.com>
Date:   Wed Apr 30 17:40:40 2014 +0100

    Merge branch 'master' of git.assembla.com:museum-guide

commit 2cb62d663ebc782cc69e5bc36a8cf22bd7d2bda3
Author: Steve Square <steven.square@btinternet.com>
Date:   Wed Apr 30 17:39:27 2014 +0100

    Added additional comments, to aid readability.

commit 9972568b65acb92f0b7b4a8324d094aba3263ad1
Author: nath247 <nathan957@hotmail.com>
Date:   Tue Apr 29 20:02:17 2014 +0100

    Tidied up display code

commit 9d22aff261903d31033e1f96e16e0c6d395692d3
Author: nath247 <nathan957@hotmail.com>
Date:   Tue Apr 29 19:26:49 2014 +0100

    Displays 1234 on the 7 segement displays

commit e77c6a21d19aa1a3ba0a32574faa5073cca2ed1e
Author: nath247 <nathan957@hotmail.com>
Date:   Tue Apr 29 16:11:35 2014 +0100

    Initial keypad release, code needs a fair bit of re-work

commit 6cec63df6bd092ff5c57252d7f72154833eac387
Author: mark randall <mark5.randall@live.uwe.ac.uk>
Date:   Mon Apr 28 14:39:34 2014 +0100

    Updated Client with all message response logic.\nNeeds keypad module, and Socket
reattempt code

commit e6fa2929092b0730451896a5a082a3614ae4a09f
Author: Steve Square <steven.square@btinternet.com>
Date:   Fri Apr 25 12:19:49 2014 +0100
```

```
    Added comments to allow easy reading

commit 99712236c0bb4c88d2372b83a3fb2b8c1af549fb
Author: Steve Square <steven.square@btinternet.com>
Date:   Fri Apr 25 12:08:09 2014 +0100

    changed user details

commit dc579ed72b77e7d5cd1ba1952042f9d136e6effa
Author: Steve Square <steven.square@btinternet.com>
Date:   Fri Apr 25 12:05:20 2014 +0100

    Add server.py:- Basic python threaded server with ssl and plain connection types.

commit 1d239e7a2df5eae46e8738cff862aa0121d2be0d
Author: mark randall <mark_the_mosquito@hotmail.com>
Date:   Wed Mar 5 23:02:01 2014 +0000

    Fixed an issue when changing from int checking, (for space/CRLF found) to bool

commit 04740b3fa578b0bcc26354c1ba50370a9958da66
Author: mark randall <mark_the_mosquito@hotmail.com>
Date:   Wed Mar 5 22:52:07 2014 +0000

    Updated include.h with regex

commit 072d5c8e254ff3facac535a9926297ad7c7fef89
Author: mark randall <mark_the_mosquito@hotmail.com>
Date:   Wed Mar 5 22:49:59 2014 +0000

    Updated Server.cpp with new parser. This should allow the new request format to be
accepted, and is slightly                                more robust. However it is
currently untested, and still needs MAC address parsing to be added (confirm MAC
format with Sami)

commit db2d7868395cd267a822e10f32cc7ccfa9b8616c
Author: mark randall <mark_the_mosquito@hotmail.com>
Date:   Thu Feb 27 16:24:00 2014 +0000

    Fixed a bug in parseRequestMsg() which was causing incorrect logic throughout the
rest of the parsing/message checking functions.

    We now default to send 0 as code, "" for title, and "" for message, if the
function finds now faults in the format of the request.

    We then check the actual command, code and protocol in checkMessages()

    TODO: 2 bugs still exist in parsing, which should be fixed after the prototype
stage.

    Bug1: We detect we get an empty line, by searching for "\r\n\r\n". Only one empty
line needs to exist, and the logic works for making sure only 1 exists... However if
```

```
the message has a \n, or a \r singularly, then we have no current method to check for
this, and error appropriately.

    e.g: This request would be valid in the server program:
    "POST HTTP/1.1\r\n\r\n#4242\n"

    Bug2: The current functionality of the string to int converter would not error if
a string/char is found within the code. Instead it will take all valid numbers before
the occurance of an invalid char/string.

    e.g. This request would be valid in the server program:
    "POST HTTP/1.1\r\n\r\n#42a42"
    The server would respond to this message, exactly the same, as if it were to get
this message:
    "POST HTTP/1.1\r\n\r\n#42"
    The latter being valid.

    These should be fixed for the final version.

commit 3464d22baa77c632425025dda97af350ebe1f770
Author: mark randall <mark_the_mosquito@hotmail.com>
Date:   Wed Feb 26 17:11:37 2014 +0000

    Updated the "if(statusMsg.getCode() != 0)" in readmessage() to correctly error
handle:

    i.e.    take the response message created by parseRequestMsg(), when we get an
invalid message,

      and send it back to the client (i.e. client sent invalid request, send a 400
response)

    We do not want to pass parsedMsg object through checkMessages() in this
circumstance (which was happening in v0.1.1) as the message would not be parsed
correctly, as it couldn't be!

commit 7e3bb8b449035dfd248266442feb2c79579a4c58
Author: mark randall <mark_the_mosquito@hotmail.com>
Date:   Wed Feb 26 17:04:45 2014 +0000

    Removed unnecessary messaging for debug purposes

commit f8fc240456faf2417e75e6cfe4dcc99531d9b2d3
Author: mark randall <mark_the_mosquito@hotmail.com>
Date:   Wed Feb 26 16:06:00 2014 +0000

    First addition for Sami's socket code, and Mark's parser for client and server -
Prototype version

commit b58c1c5e0a85efe66235c1a3a7ef45d9544205ea
Author: Steve Square <steven.square@btinternet.com>
Date:   Thu Feb 6 22:45:36 2014 +0000

    Fixed dynamic linking system. now able to link all elements in linked list.
```

```
commit 0bd164be765552ae6542f51bb5f1209b3ae3837d
Author: Steve Square <steven.square@btinternet.com>
Date:   Thu Feb 6 19:22:15 2014 +0000

    Significant changes to elemnt behaviour( Work in Progress.)

commit fe7fbb9396aba4821bccdc3f27d1509a2de7e34b
Author: Steve Square <steven.square@btinternet.com>
Date:   Sat Dec 21 14:18:10 2013 +0000

    Minor changes to list functions.

commit 7e579f3c2784aa3ccddef2d0ccee5cc1274292d1
Author: Steve Square <steven.square@btinternet.com>
Date:   Fri Dec 20 11:19:26 2013 +0000

    fixed segv.

commit a63daa67c375a17a9a0729da5a7f29e34d9c5e99
Author: Steve Square <steven.square@btinternet.com>
Date:   Thu Dec 19 21:39:59 2013 +0000

    Added linked list traversal, bug causing segv due to incorrect memory access after
printing elements, best not to at present.

commit 87d00c6549c79d32aeebaadcb6fbe0f2c1ee18ec
Author: Steve Square <steven.square@btinternet.com>
Date:   Tue Dec 17 00:30:47 2013 +0000

    Removed 1, compile.log, gstaudio, gstelementlist.h, gstelementlist.h~ as they are
null and void now

commit 550f7dfb6415387a748dc4bdaad2e3eb167ece11
Author: Steve Square <steven.square@btinternet.com>
Date:   Tue Dec 17 00:26:55 2013 +0000

    Add element creation functionality. improved code style

commit 1d43060ef44569aca21ef1262400d95109a925a1
Author: Steve Square <steven.square@btinternet.com>
Date:   Mon Dec 16 17:59:10 2013 +0000

    Started work on gstreamer audio integration. simple at present, only displays
version.

commit 481f20115df70649ec6ab4c5390db510f29df164
Author: ssquare <steven.square@btinternet.com>
Date:   Mon Nov 4 21:52:46 2013 +0000

    Project Kickoff
```

Makefiles

Throughout the project, we have used what are called makefiles. As the software project consists of many source codes, we can have complex and long compiler commands. Some of these compiler options may be so rarely used that they are hard to remember. Through using makefiles we have been able to maintain a consistent development environment. Make is used to carry out the compilation and linkage operations. This ensures accuracy, consistency and also records all the file names and commands for future reference and repetition.

An example of a makefile that we have used in the project is given below:

```
CC = g++
CFLAGS = -Wall -g
LIBS =
INCLUDES=
TARGET = museumguide
SRCS = main.cpp keypad.c display.c gstaudio.cpp Client.cpp Server.cpp
OBJS = $(SRCS:.c=.o)

.PHONY: depend clean

all: $(TARGET)
      @echo Remember make final.

$(TARGET): $(OBJS)
      $(CC) $(CFLAGS) $(INCLUDES) -o $(TARGET) $(OBJS) $(LIBS)

.c.o:
      $(CC) $(CFLAGS) -D _NO_LCD_ -D _NO_KEYPAD_ $(INCLUDES) -c $< -o $@

final: $(OBJS)
      $(CC) $(CFLAGS) $(INCLUDES) lcd.c lcd.h -o $(TARGET) $(OBJS) $(LIBS)

clean:
      $(RM) *.o *~ $(MAIN)

depend: $(SRCS)
      makedepend $(INCLUDES) $^
```