**Original project plan and Gantt chart up to interim report submission**

| Tasks | Start Date | Duration (Days) | End Date |
|---|---|---|---|
| Define scope of the system | 11/2/2014 | 3 | 11/4/2014 |
| Prepare questionnaires for surveys | 11/5/2014 | 5 | 11/9/2014 |
| Conduct interviews | 11/10/2014 | 8 | 11/17/2014 |
| Study similar systems | 11/18/2014 | 5 | 11/22/2014 |
| Analysis of the result | 11/23/2014 | 5 | 11/27/2014 |
| Prepare SRS document | 11/28/2014 | 16 | 12/13/2014 |
| Design system architecture | 12/14/2014 | 15 | 12/28/2014 |
| Create ER diagram for the system | 12/29/2014 | 7 | 1/4/2015 |

Table showing various tasks, start date and end date of the old project plan
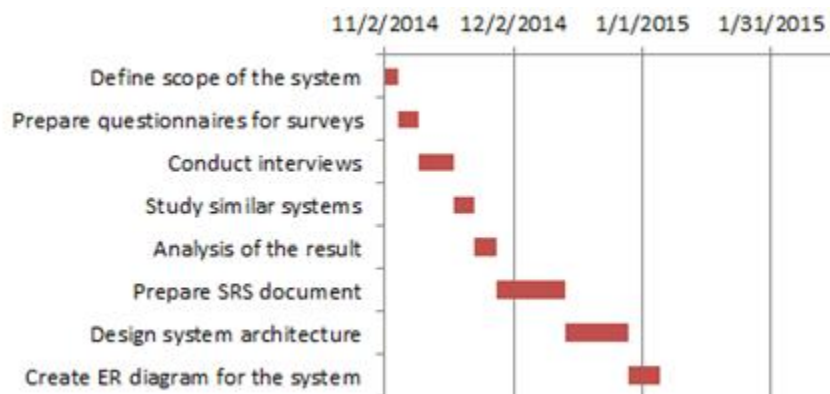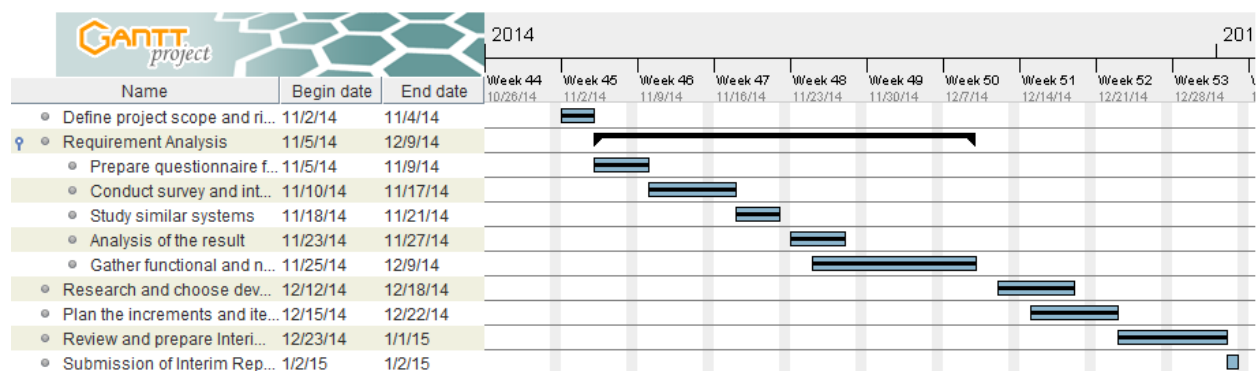


Gantt chart for the original project plan

**Actual project plan and Gantt chart up to interim report submission**

| Work Package | Activity | Begin date | End date | Duration (days) |
|---|---|---|---|---|
| Project initiation | Define project scope and risks | 11/2/2014 | 11/4/2014 | 3 |
| | Prepare questionnaire for survey | 11/5/2014 | 11/9/2014 | 4 |
| Requirement | Conduct survey and interviews | 11/10/2014 | 11/17/2014 | 7 |

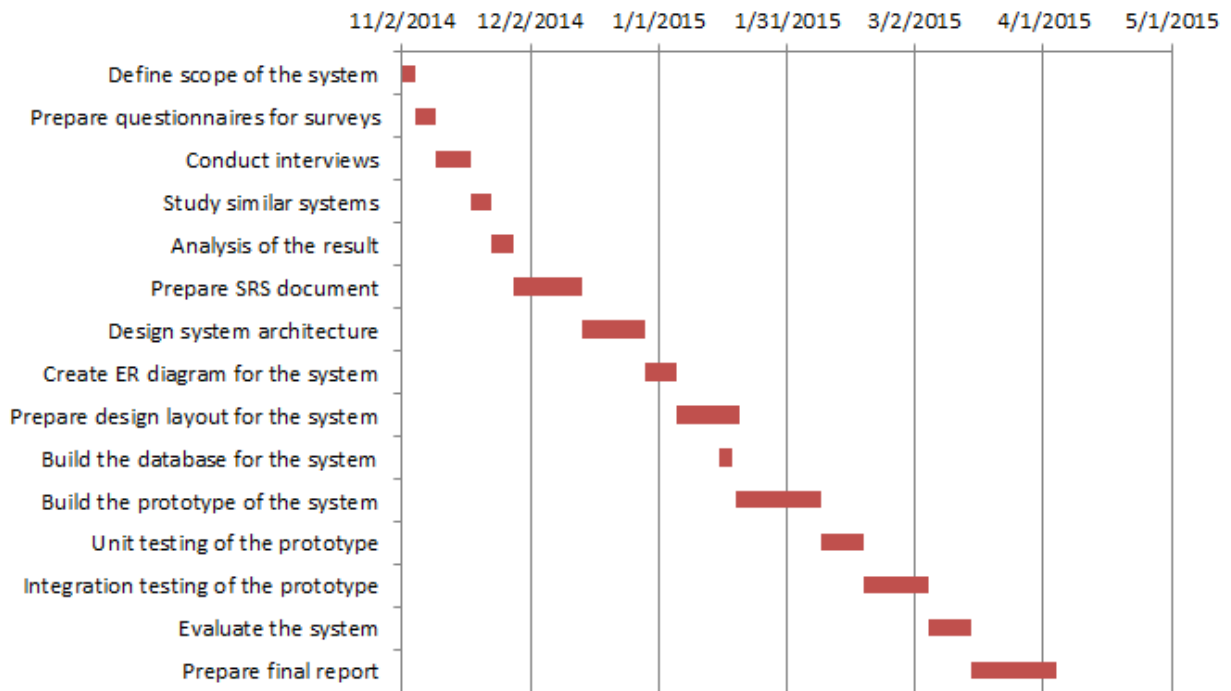| Analysis | Study similar systems | 11/18/2014 | 11/21/2014 | 4 |
|---|---|---|---|---|
| | Analysis of the result | 11/23/2014 | 11/27/2014 | 5 |
| | Gather functional and non-functional requirements | 11/25/2014 | 12/9/2014 | 13 |
| | Research and choose development methodology | 12/12/2014 | 12/18/2014 | 6 |
| | Plan the increments and iterations | 12/15/2014 | 12/22/2014 | 7 |
| | Review and prepare Interim Report | 12/23/2014 | 1/1/2015 | 9 |
| | Submission of Interim Report | 1/2/2015 | 1/2/2015 | 1 |

Table showing various activities along with the start date and end date



Gantt chart for the actual project plan

**Complete project plan and Gantt chart for the FYP**

**(Initial)**

| | 11/2/2014 | 12/2/2014 | 1/1/2015 | 1/31/2015 | 3/2/2015 | 4/1/2015 | 5/1/2015 |
|---|---|---|---|---|---|---|---|

Define scope of the system

Prepare questionnaires for surveys

Conduct interviews

Study similar systems

Analysis of the result

Prepare SRS document

Design system architecture

Create ER diagram for the system

Prepare design layout for the system

Build the database for the system

Build the prototype of the system

Unit testing of the prototype

Integration testing of the prototype

Evaluate the system

Prepare final report

**Revised Gantt chart**

| Name | Begin date | End date |
|---|---|---|
| Define project scope and risks | 11/2/14 | 11/4/14 |
| Requirement Analysis | 11/5/14 | 12/9/14 |
| Prepare questionnaire for survey | 11/5/14 | 11/9/14 |
| Conduct survey and interviews | 11/10/14 | 11/17/14 |
| Study similar systems | 11/18/14 | 11/21/14 |
| Analysis of the result | 11/23/14 | 11/27/14 |
| Gather functional and non-functional req... | 11/25/14 | 11/29/14 |
| Research and choose development method... | 12/12/14 | 12/18/14 |
| Plan the increments and iterations | 12/15/14 | 12/22/14 |
| Review and prepare Interim Report | 12/23/14 | 1/1/15 |
| Submission of Interim Report | 1/2/15 | 1/2/15 |
| Create an ER diagram for the whole system | 1/4/15 | 1/9/15 |
| Design system architecture | 1/11/15 | 1/20/15 |
| Plan and schedule increments | 1/21/15 | 1/26/15 |
| Increment 1 (Login and Register) | 1/27/15 | 2/1/15 |
| Design the frontend for the login and reg... | 1/27/15 | 2/1/15 |
| Develop the increment | 2/2/15 | 2/4/15 |
| Test and review the increment | 2/5/15 | 2/8/15 |
| Increment 2 (Service post, order and rate) | 2/9/15 | 2/23/15 |
| Design the frontend for the increment | 2/9/15 | 2/13/15 |
| Develop and integrate the increment | 2/15/15 | 2/19/15 |
| Test and review the increment | 2/20/15 | 2/23/15 |
| Increment 3 (Accept/Reject order and payme... | 2/24/15 | 3/3/15 |
| Design the frontend | 2/24/15 | 2/25/15 |
| Develop and integrate the increment | 2/26/15 | 3/1/15 |
| Test and review the increment | 3/1/15 | 3/3/15 |
| Increment 4 (Messaging and search posts) | 3/4/15 | 3/9/15 |
| Design the frontend | 3/4/15 | 3/4/15 |
| Develop and integrate the increment | 3/5/15 | 3/6/15 |
| Test and review the increment | 3/8/15 | 3/9/15 |
| Increment 5 (File upload and payment witho... | 3/10/15 | 3/16/15 |
| Design the frontend | 3/10/15 | 3/11/15 |
| Develop and integrate the increment | 3/12/15 | 3/13/15 |
| Test and review the increment | 3/15/15 | 3/16/15 |
| Increment 6 (Admin module) | 3/17/15 | 3/25/15 |
| Design the frontend for admin | 3/17/15 | 3/19/15 |
| Develop and integrate module | 3/20/15 | 3/23/15 |
| Test and review the increment | 3/24/15 | 3/25/15 |
| Prepare Final Report for FYP | 3/26/15 | 4/17/15 |

**Entity Relationship Diagram**

**Unnormalized form**

USERS ( user_id, fname, lname, gender, dob, display_name, address, about, profilePic, password, created_at, active, authKey, email, verified, accessToken, {{{ post_id, title, description, view_count, {rating, user_id, datetimestamp}, price, currency, image_url, expiry_date, featured, datetimestamp, max_active_order, max_delivery_days, active, slug, {{order_id, orderd_by, details, status, datetimestamp, type, rejected_reason, rejected_date, accepted_date, status, payment, delivery_date, complete_request, complete_request_date, cancel_request, cancel_request_date, closed_date, transaction_id, post_price, currency, amount, payment_status, payer_email, datetimestamp, error_log, withdraw_id, user_id, request_date, complete, datetimestamp}, status_id, user_id, status, datetimestamp, file_url}}, category_id, category_name, created_date }},{report_id, reported_by, report, datetimestamp, active}, {message_id, subject, message, thread_id, datetimestamp, read_m, status, from_id},{comment_id, comment, rating, datetimestamp, active, comment_by}, {notification_id, notification, datetimestamp, read, source, type, status, post_id})

**1NF**

users (user_id, fname, lname, gender, dob, display_name, address, about, profilePic, password, created_at, active, authKey, email, verified, accessToken)

post_categories (category_id, category_name, created_date, admin_id*)

post_services (post_id, user_id*, category_id*, title, description, view_count, price, currency, image_url, expiry_date, featured, datetimestamp, max_active_order, max_delivery_days, active, slug)

post_ratings (post_id, user_id, rating, datetimestamp)

post_order (order_id, post_id*, ordered_by*, details, status, datetimestamp, type, rejected_reason, rejected_date, accepted_date, status, payment, delivery_date, complete_request, complete_request_date, cancel_request, cancel_request_date, closed_date, transaction_id, post_price, currency, amount, payment_status, payer_email, datetimestamp, error_log, withdraw_id, user_id*, request_date, complete, datetimestamp)

task_status (status_id, user_id*, order_id*, status, datetimestamp, file_url)

messages (<u>message_id</u>, subject, message, from_id, thread_id, datetimestamp, read_m, status, user_id*)

user_comments (<u>comment_id</u>, comment_by, comment, rating, datetimestamp, active, user_id*)

notifications (<u>notification_id</u>, notification, datetimestamp, read, source*, type, status, post_id, user_id***)**

flag_reports (<u>report_id</u>, reported_by*, user_id*, report, datetimestamp, active)

*There is no partial functional dependencies. Thus, it is also in 2NF.*

**<u>2NF</u>**
users (<u>user_id,</u> fname, lname, gender, dob, display_name, address, about, profilePic, password, created_at, active, authKey, email, verified, accessToken)

post_categories (<u>category_id</u>, category_name, created_date, admin_id*)

post_services (<u>post_id</u>, user_id*, category_id*, title, description, view_count, price, currency, image_url, expiry_date, featured, datetimestamp, max_active_order, max_delivery_days, active, slug)

post_ratings (<u>post_id, user_id</u>, rating, datetimestamp)

post_order (<u>order_id,</u> post_id*, ordered_by*, details, status, datetimestamp, type, rejected_reason, rejected_date, accepted_date, status, payment, delivery_date, complete_request, complete_request_date, cancel_request, cancel_request_date, closed_date, transaction_id, post_price, currency, amount, payment_status, payer_email, datetimestamp, error_log, withdraw_id, user_id*, request_date, complete, datetimestamp)

task_status (<u>status_id</u>, user_id*, order_id*, status, datetimestamp, file_url)

messages (<u>message_id</u>, subject, message, from_id, thread_id, datetimestamp, read_m, status, user_id*)

user_comments (<u>comment_id</u>, comment_by, comment, rating, datetimestamp, active, user_id*)

notifications (<u>notification_id</u>, notification, datetimestamp, read, source*, type, status, post_id, user_id***)**

flag_reports (<u>report_id</u>, reported_by*, user_id*, report, datetimestamp, active)


## 3NF

*Transitive Dependencies*
**post_orders**
transaction_id -> post_price, currency, amount, payment_status, payer_email, datetimestamp, error_log

withdraw_id -> transaction_id, user_id, request_date, complete, datetimestamp


users (<u>user_id,</u> fname, lname, gender, dob, display_name, address, about, profilePic, password, created_at, active, authKey, email, verified, accessToken)

post_categories (<u>category_id</u>, category_name, created_date, admin_id*)

post_services (<u>post_id</u>, user_id*, category_id*, title, description, view_count, price, currency, image_url, expiry_date, featured, datetimestamp, max_active_order, max_delivery_days, active, slug)

post_ratings (<u>post_id, user_id</u>, rating, datetimestamp)

post_orders (<u>order_id,</u> post_id*, ordered_by*, details, status, datetimestamp, type, rejected_reason, rejected_date, accepted_date, status, payment, delivery_date, complete_request, complete_request_date, cancel_request, cancel_request_date, closed_date)

task_status (<u>status_id</u>, user_id*, order_id*, status, datetimestamp, file_url)

messages (<u>message_id</u>, subject, message, from_id, thread_id, datetimestamp, read_m, status, user_id*)

user_comments (<u>comment_id</u>, comment_by, comment, rating, datetimestamp, active, user_id*)

notifications (<u>notification_id</u>, notification, datetimestamp, read, source*, type, status, post_id, user_id*)

flag_reports (<u>report_id</u>, reported_by*, user_id*, report, datetimestamp, active)

transaction (<u>transaction_id</u>, order_id*, post_price, currency, amount, payment_status, payer_email, datetimestamp, error_log)

transaction_withdraw (<u>withdraw_id,</u> transaction_id*, user_id*, request_date, complete, datetimestamp)

*Separating child entities from parent entities,*
**post_services -> post_views**
**post_order -> accepted_orders, rejected_orders**
**users -> admin**
**task_status -> task_files**

**FINAL SET OF NORMALIZED TABLES**

users (<u>user_id,</u> fname, lname, gender, dob, display_name, address, about, profilePic, password, created_at, active, authKey, email, verified, accessToken)

admin (<u>user_id</u>)

post_categories (<u>category_id</u>, category_name, created_date, admin_id*)

post_services (<u>post_id</u>, user_id*, category_id*, title, description, price, currency, image_url, expiry_date, featured, datetimestamp, max_active_order, max_delivery_days, active, slug)

post_views (post_id, view_count)

post_ratings (<u>post_id, user_id</u>, rating, datetimestamp)

post_orders (<u>order_id</u>, post_id*, ordered_by*, details, status, datetimestamp, type)

accepted_orders (<u>order_id</u>, accepted_date, status, payment, delivery_date, complete_request, complete_request_date, cancel_request, cancel_request_date, closed_date)

rejected_orders (<u>order_id</u>, rejected_reason, rejected_date)

task_status (<u>status_id</u>, user_id*, order_id*, status, datetimestamp, file_url)

task_files (<u>status_id</u>, file_url)

messages (<u>message_id</u>, subject, message, from_id, thread_id, datetimestamp, read_m, status, user_id*)

user_comments (<u>comment_id</u>, comment_by, comment, rating, datetimestamp, active, user_id*)

notifications (<u>notification_id</u>, notification, datetimestamp, read, source*, type, status, post_id, user_id*)

flag_reports (<u>report_id</u>, reported_by*, user_id*, report, datetimestamp, active)

transaction (<u>transaction_id</u>, order_id*, post_price, currency, amount, payment_status, payer_email, datetimestamp, error_log)

transaction_withdraw (<u>withdraw_id,</u> transaction_id*, user_id*, request_date, complete, datetimestamp)
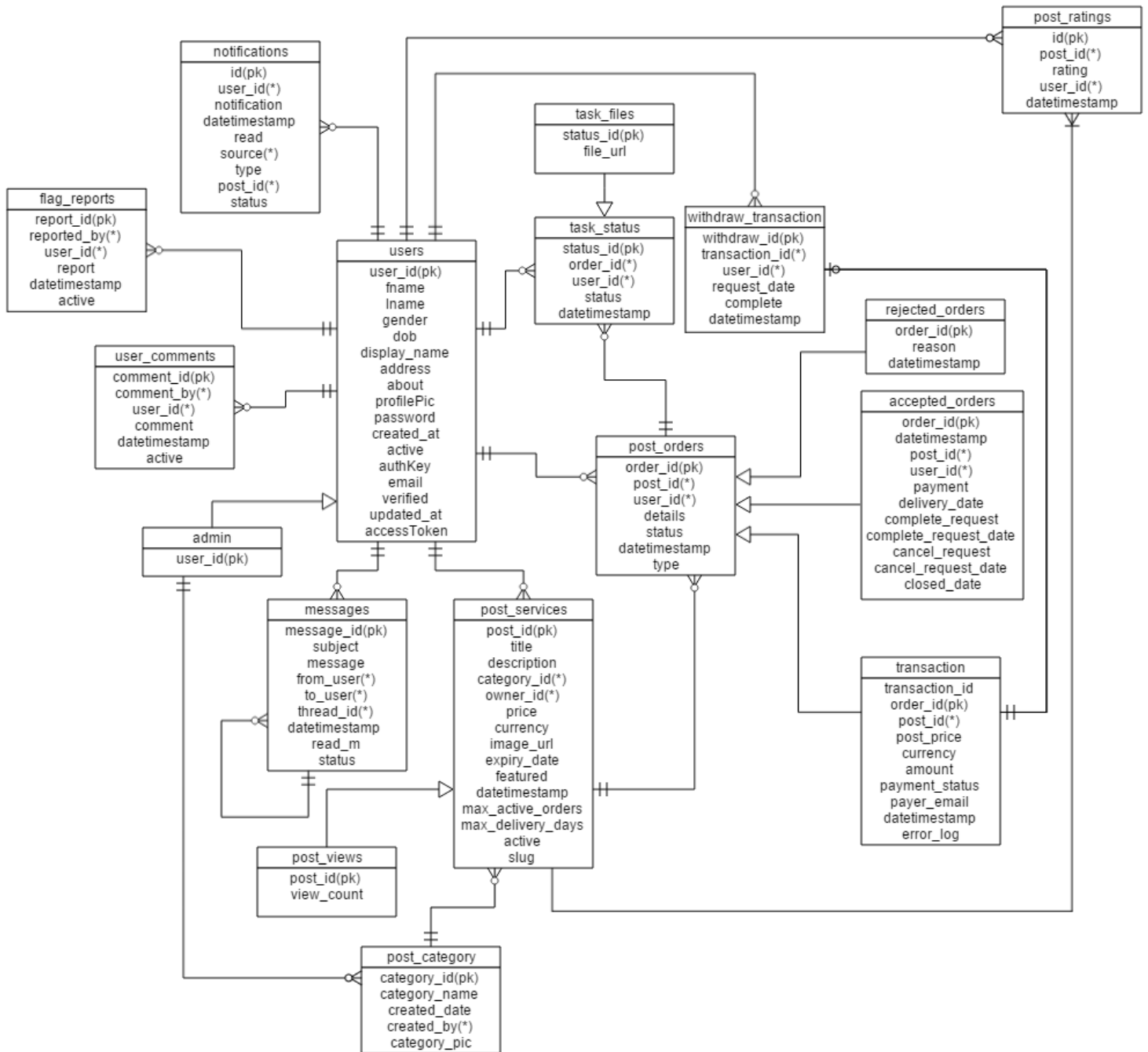
**Entity Relationship Diagram**



Fig: Entity relationship diagram showing various entities, attributes and relationship