

References

- [1] K. ZICKUHR, "Location Based Services," PewResearchCenter, 12 9 2013. [Online]. Available: <http://www.pewinternet.org/2013/09/12/location-based-services/>. [Accessed 21 11 2015].
- [2] H. Bray, 30 5 2014. [Online]. Available: <http://blogs.discovermagazine.com/crux/2014/04/30/how-location-based-apps-will-shape-the-future-of-shopping/#.Vk9KZ3YrLDd>. [Accessed 24 8 2015].
- [3] A. Grill, "londoncalling.co," [Online]. Available: <http://londoncalling.co/2008/05/location-based-advertising-introduction/>. [Accessed 10 11 2015].
- [4] E. Corporation, "Class 5," Ebay Pvt(Ltd), [Online]. Available: <https://www.close5.com/>. [Accessed 3 9 2015].
- [5] D. Olandff, 12 7 2012. [Online]. Available: <http://thenextweb.com/apps/2012/07/05/yardsale-for-ios-might-be-the-quickest-way-to-sell-all-of-the-stuff-sitting-around-your-house/>. [Accessed 11 8 2015].
- [6] A. Popescu, "W3C," Google, Inc, 11 7 2014. [Online]. Available: <http://dev.w3.org/geo/api/spec-source.html>. [Accessed 12 8 2015].
- [7] "W3Schools.com," W3Schools.com, [Online]. Available: http://www.w3schools.com/html/html5_geolocation.asp. [Accessed 12 9 2015].
- [8] G. Inc, "Google Maps Directions API," Google Inc, [Online]. Available: <https://developers.google.com/maps/documentation/directions/intro>. [Accessed 21 11 2015].
- [9] ikman.lk, "ikman.lk," ikman.lk, [Online]. Available: <http://ikman.lk/en/help/about>. [Accessed 20 10 2015].
- [10] O. Marketplace, "Oodle Marketplace," Oodle Marketplace, [Online]. Available: <http://www.oodle.com/info/about/>. [Accessed 8 10 2015].
- [11] H. L. Solutions, "http://hitlanka.com/," Hit Lanka Solutions, [Online]. Available: <http://hitlanka.com/web-advertising-online-buy-and-sell-best-free-classifids-website-in-sri-lanka.html>. [Accessed 8 11 2015].

- [12] marketplace.lk, "marketplace.lk," marketplace.lk, [Online]. Available: <http://marketplace.lk/en/static/index/id/3>. [Accessed 8 11 2015].
- [13] B. Jiang, "Location-based services and GIS in perspective," 2006. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.66.3566&rep=rep1&type=pdf>. [Accessed 8 11 2015].
- [14] A. Buczkowski, "<http://geoawesomeness.com/>," 2012. [Online]. Available: <http://geoawesomeness.com/knowledge-base/location-based-services/location-based-services-applications/>. [Accessed 8 11 2015].
- [15] B. Media, "<http://www.geocomm.com/>," [Online]. Available: <http://www.geocomm.com/channel/mobile/airflashsurvey/>. [Accessed 8 11 2015].
- [16] B. Bosomworth, "Mobile Marketing Statistics 2015," 22 8 2015. [Online]. Available: <http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/>. [Accessed 20 11 2015].
- [17] H. Gupta, "dzone.com," 10 7 2010. [Online]. Available: <https://dzone.com/articles/spring-framework-architecture>. [Accessed 20 11 2015].
- [18] L. T. Ben Alex, "docs.spring.io," [Online]. Available: <https://docs.spring.io/spring-security/site/docs/3.0.x/reference/springsecurity.html>. [Accessed 20 11 2015].
- [19] T. D. C. S. Oliver Gierke, "2015," 15 11. [Online]. Available: <http://docs.spring.io/spring-data/jpa/docs/current/reference/html/>. [Accessed 21 11 2015].
- [20] MD5.net, "MD5.net," MD5.net, [Online]. Available: <http://www.md5.net/>. [Accessed 07 03 2016].

Appendix

9.1 Appendix A – Sequence Diagrams

9.1.1 Login User

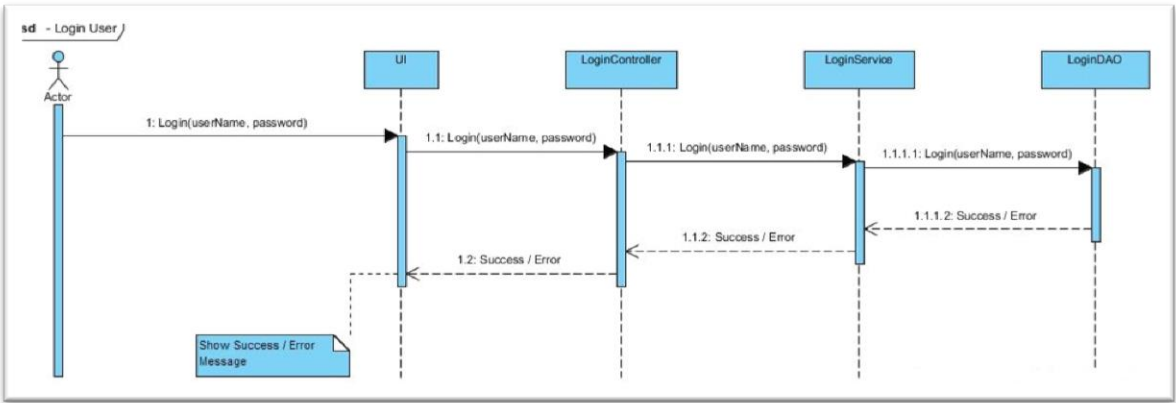


Figure 15: Login User - Sequence Diagram

9.1.2 Post Advertisement

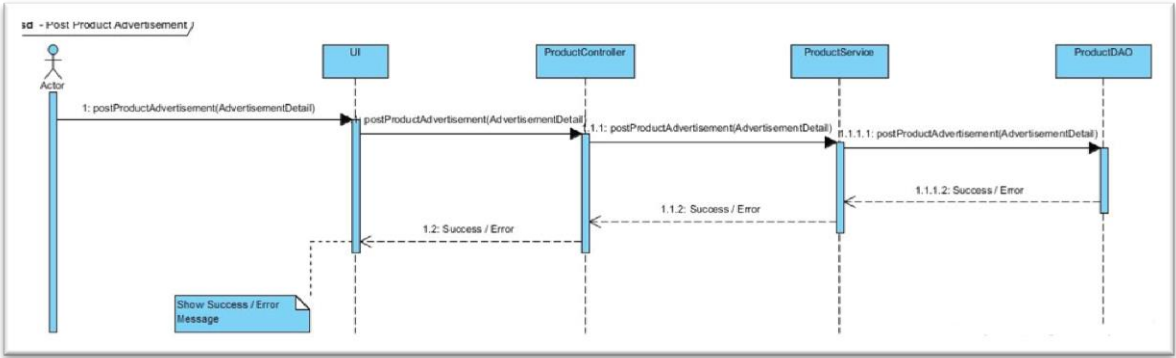


Figure 16: Post Advertisement - Sequence Diagram

9.1.3 Search Advertisement

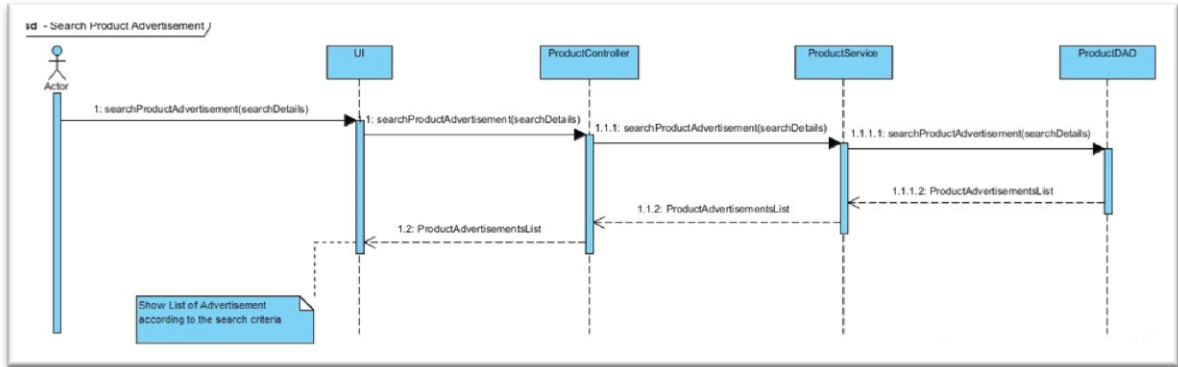


Figure 17: Search Advertisement - Sequence Diagram

9.1.4 Rank Advertisement

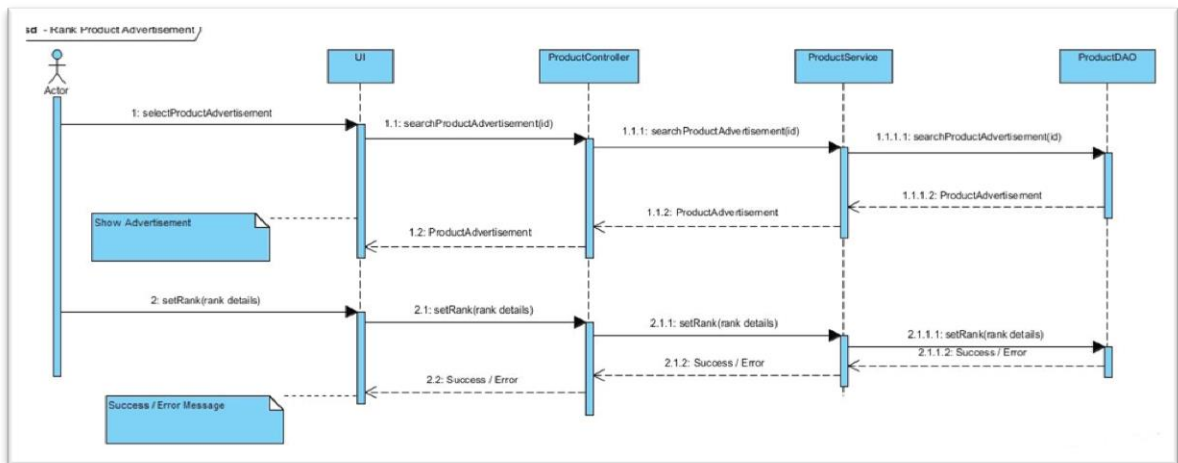


Figure 18: Rank Advertisement - Sequence Diagram

9.1.5 Comment on Advertisement

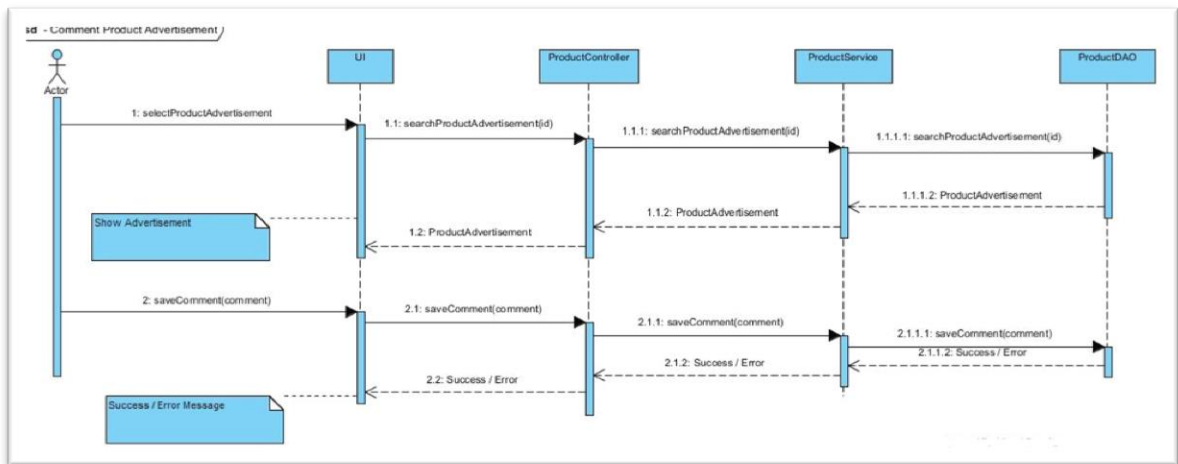


Figure 19: Comment on Advertisement - Sequence Diagram

9.1.6 Mark Favorite Advertisement

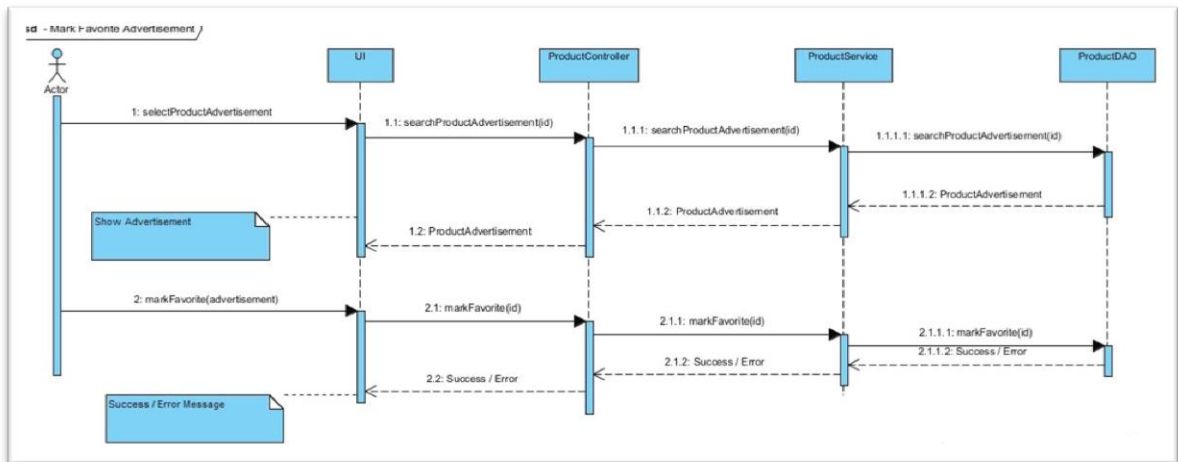


Figure 20: Mark Favorite Advertisement - Sequence Diagram

9.1.7 Send Messages

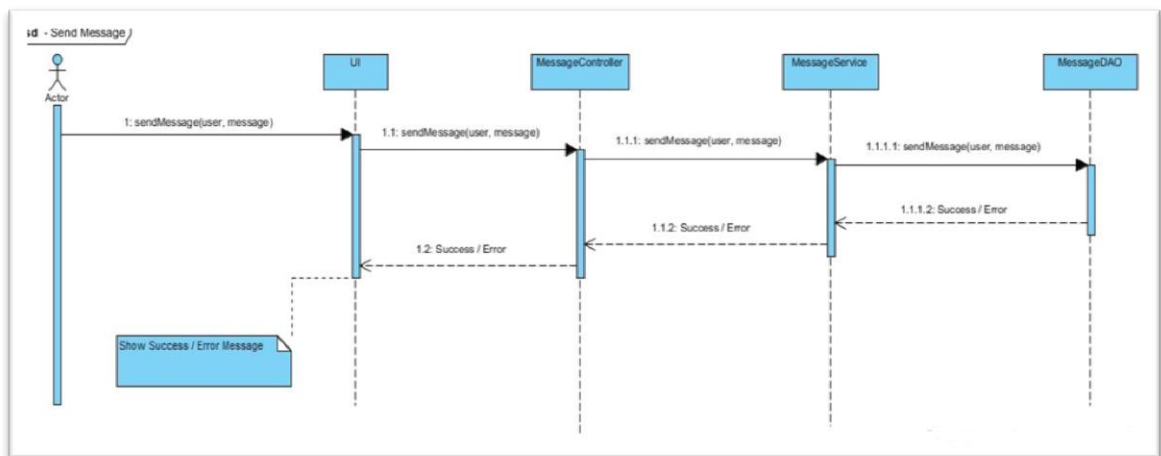


Figure 21: Send Messages - Sequence Diagram

9.2 Appendix B – Class Diagram

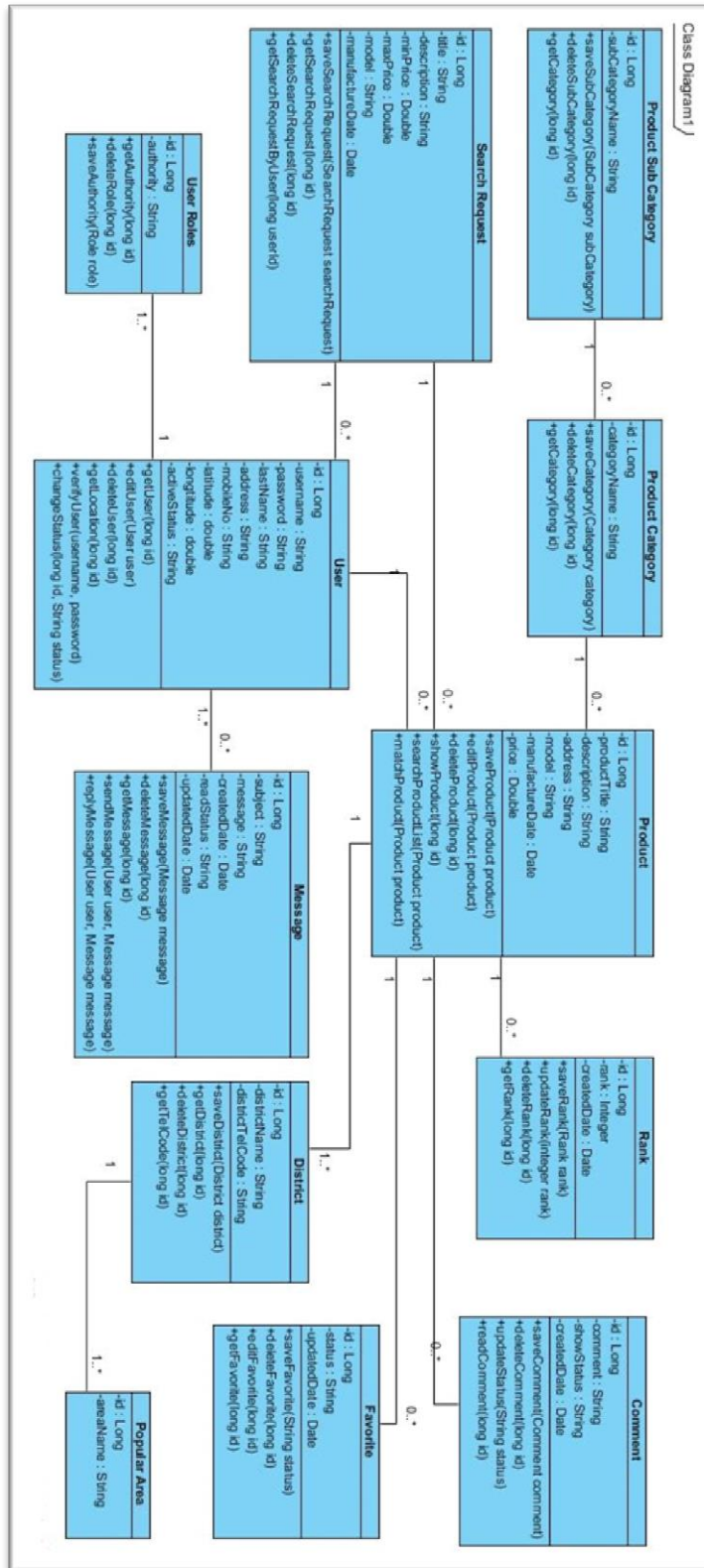


Figure 22: Class Diagram

9.3 Appendix C – ER Diagram

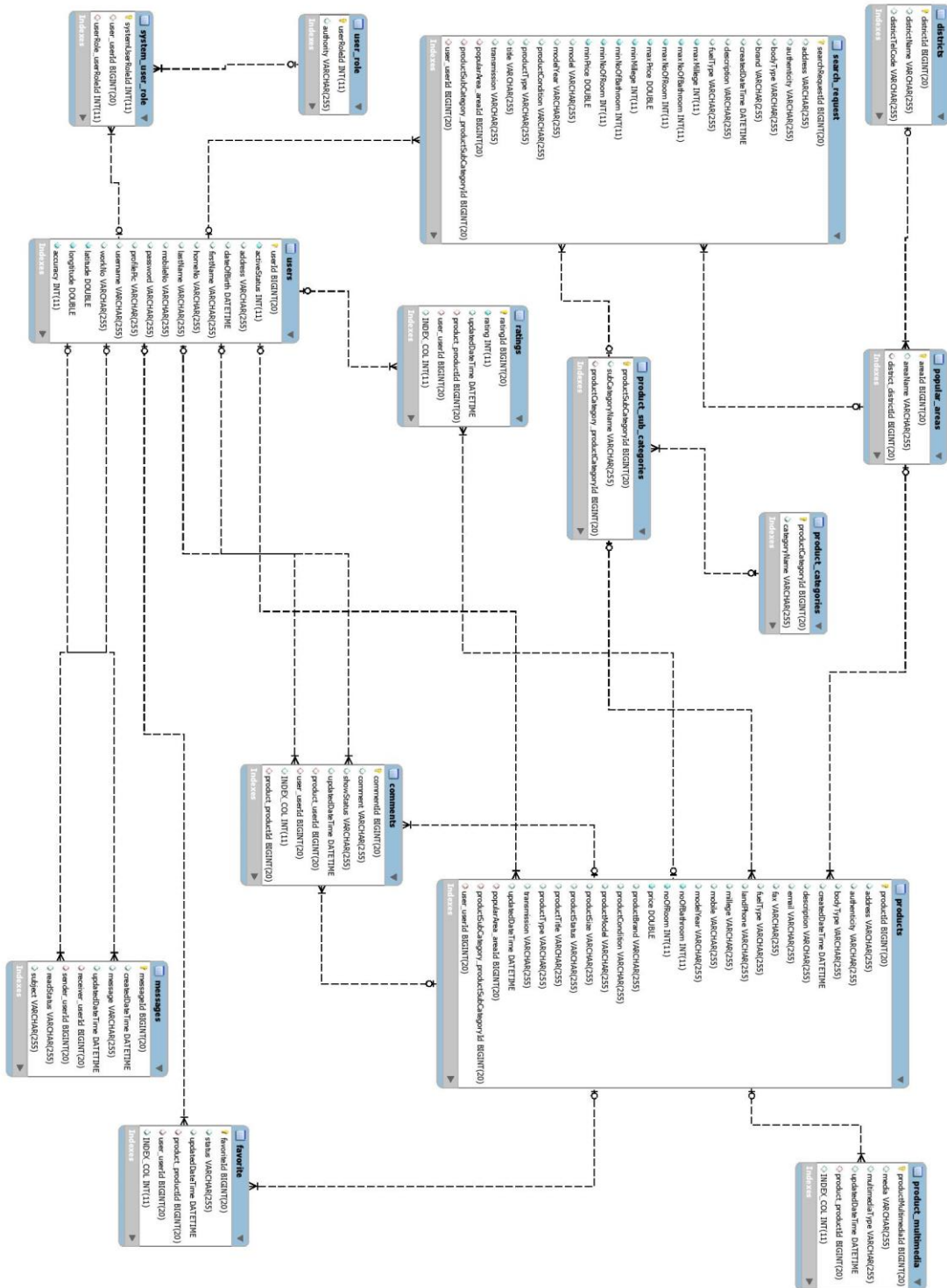


Figure 23: ER Diagram - Database Level

9.4 Appendix D – Screen Shots

9.4.1 Computer Screen Shots

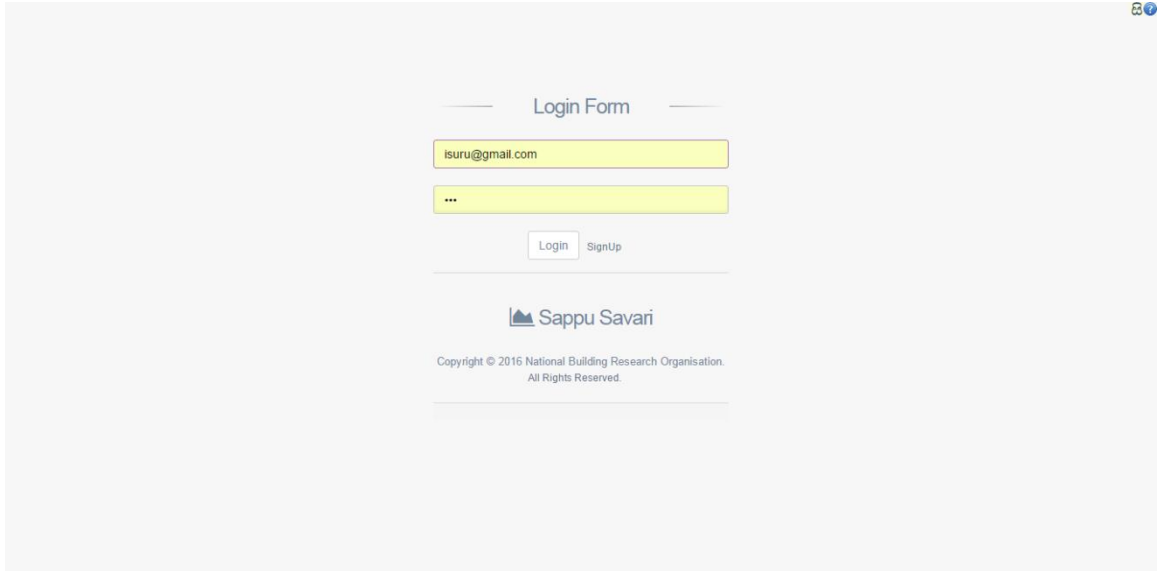


Figure 19: Login

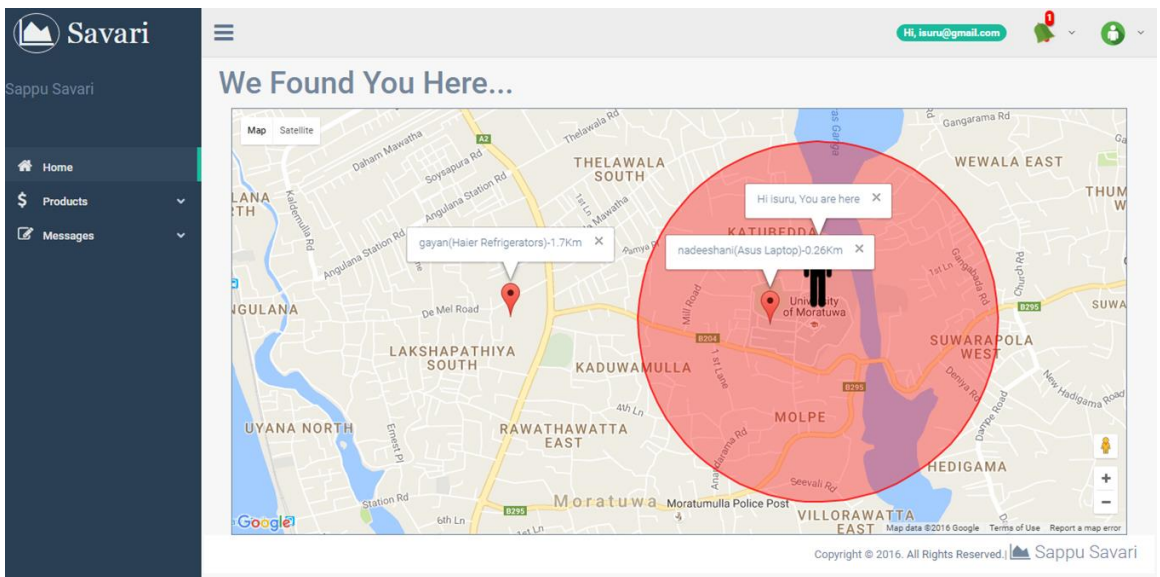


Figure 24: Home

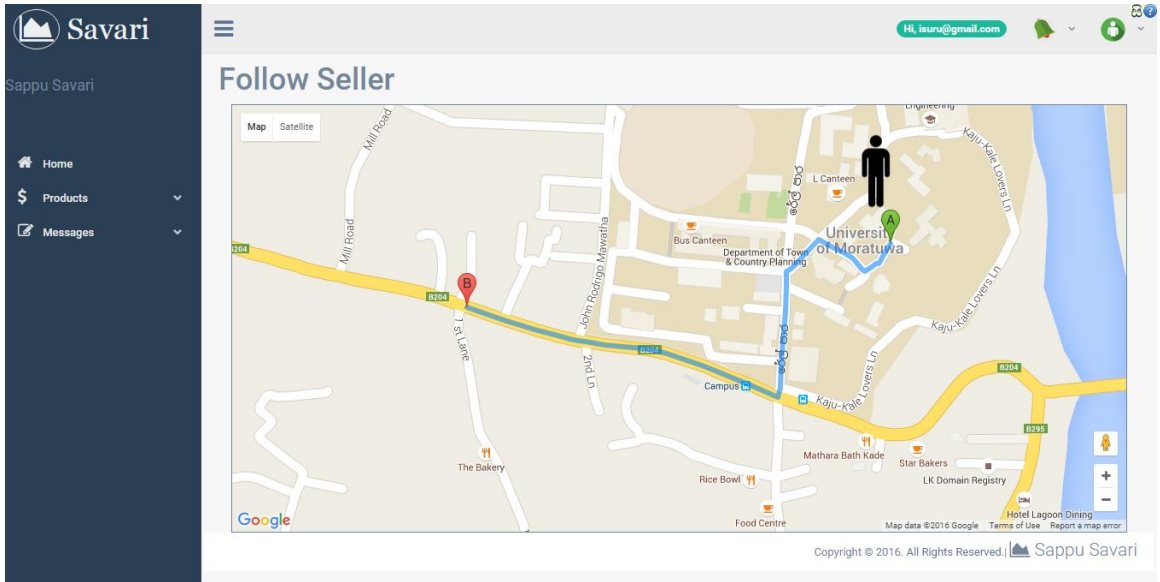


Figure 25: Follow User

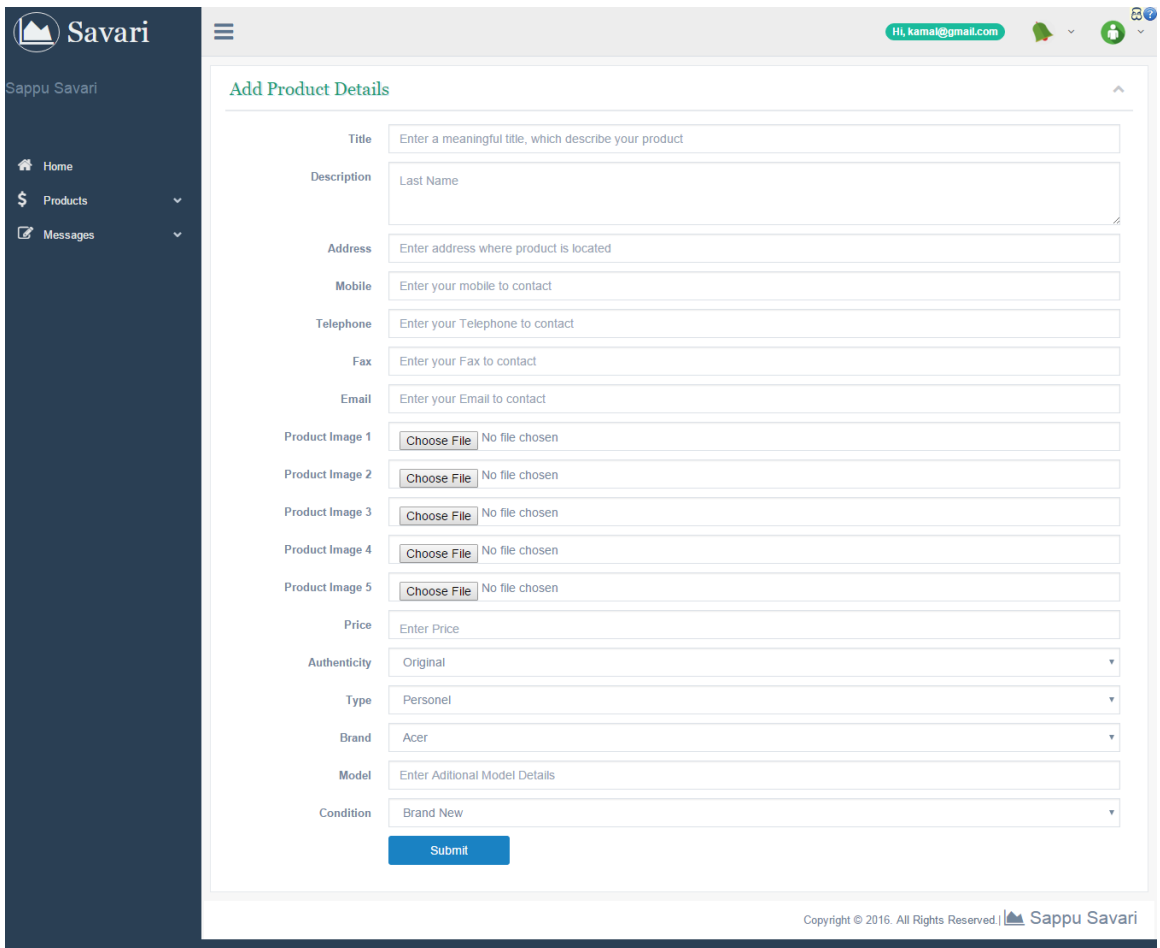


Figure 26: Add Product

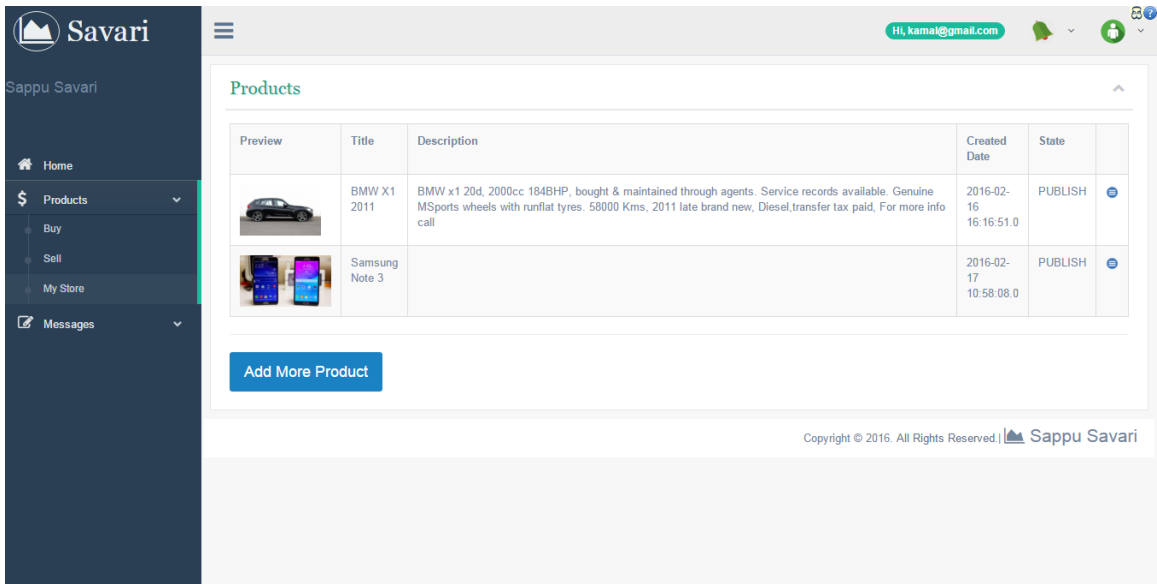


Figure 27: My Store

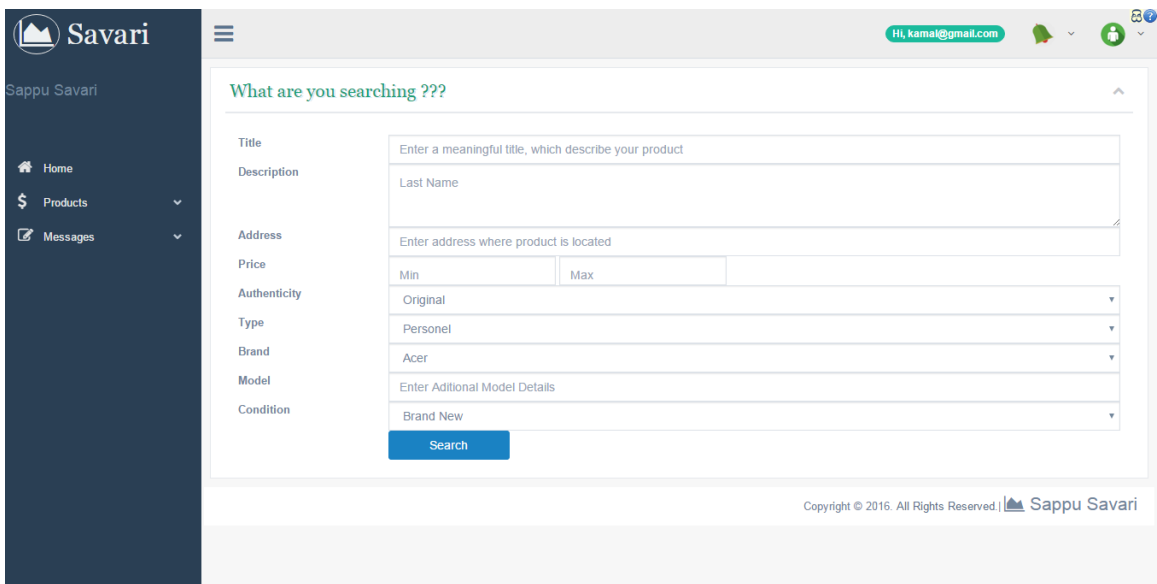


Figure 28: Search Product

Savari

Hi, kamal@gmail.com

Samsung Note 3 2016-02-17 10:58:08.0

Overall Rating : 4

Email	Mobile	Price
kamal@gmail.com	0777123456	30,000

Contact Owner Follow Seller

Rating

●●●●○

Comment Box

Please add your genuine comments regarding this product or the service. This will help other to make their decision.

This is a comment
 kamal@gmail.com(2016-03-06 20:39:15.0)

Copyright © 2016. All Rights Reserved. | Savari

Figure 29: View Product

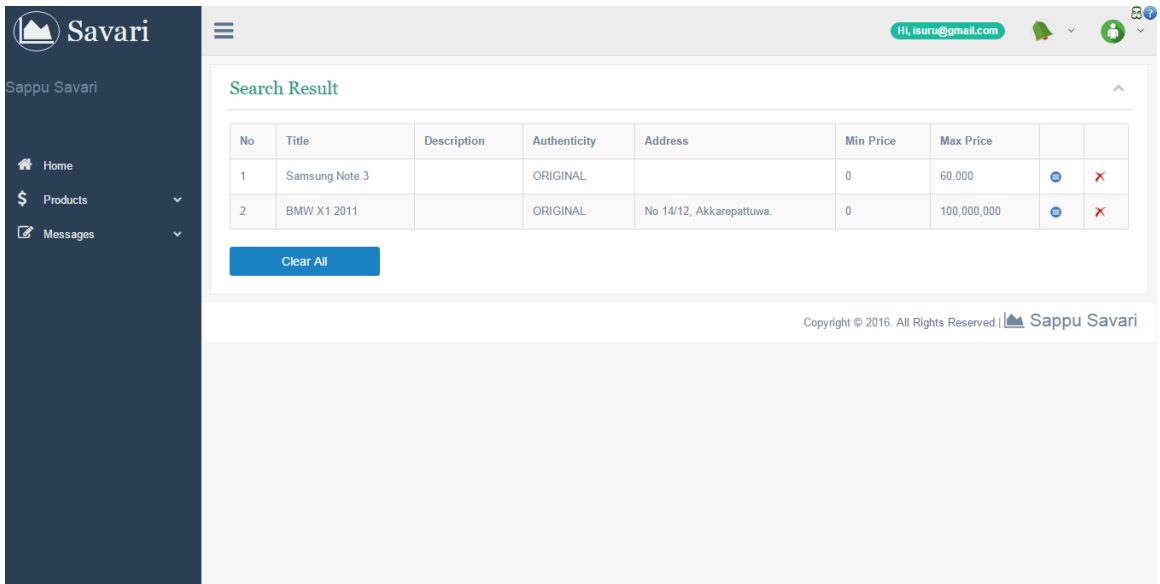


Figure 30: Search Results

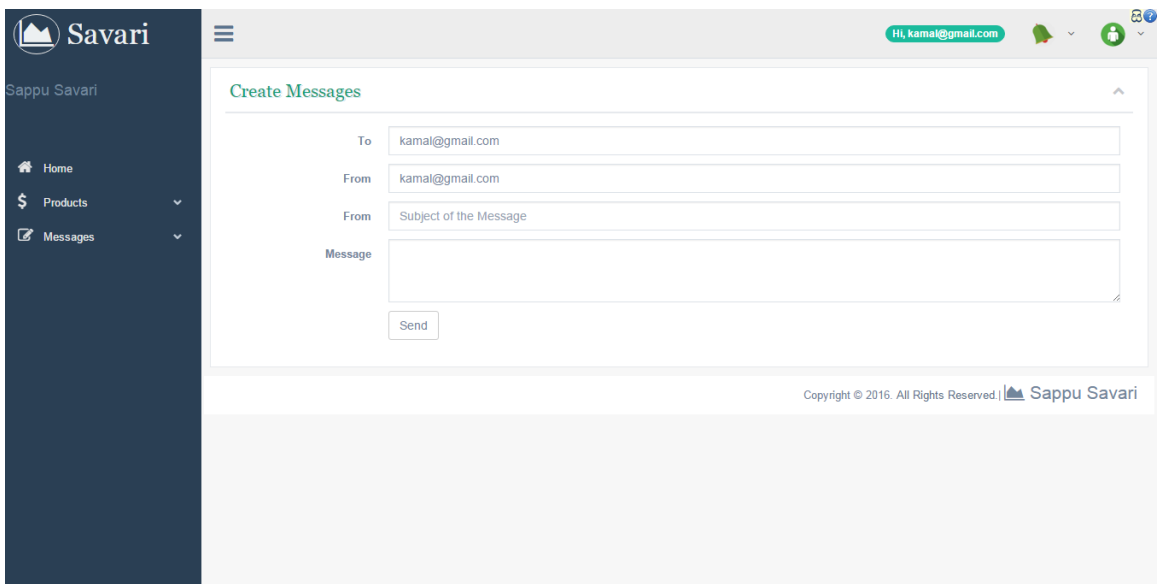


Figure 31: Send Messages

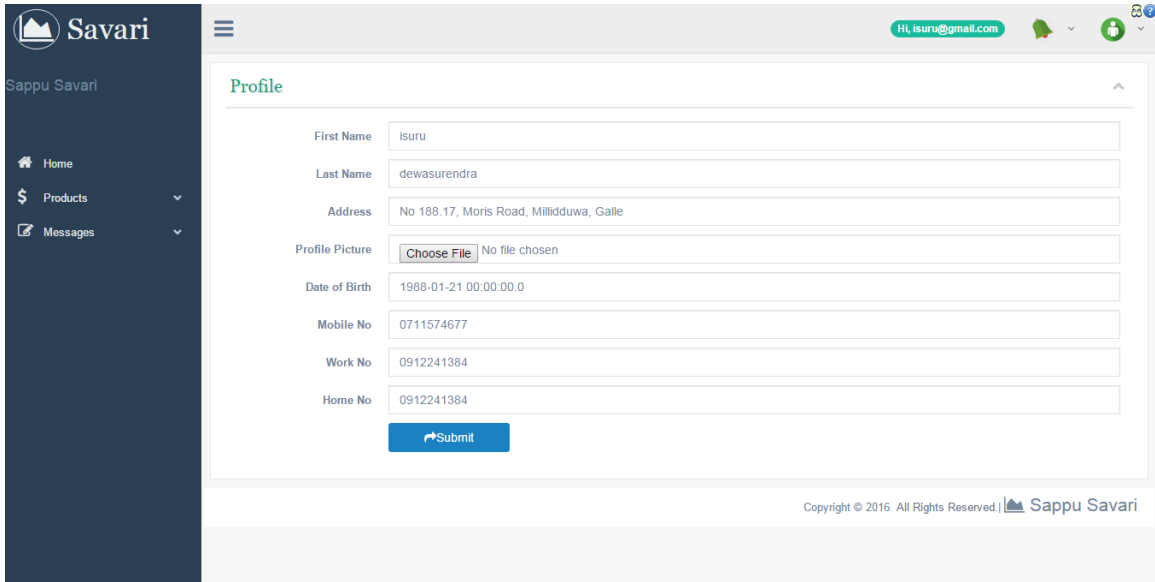


Figure 32: User Profile

9.4.2 Mobile Screen Shots

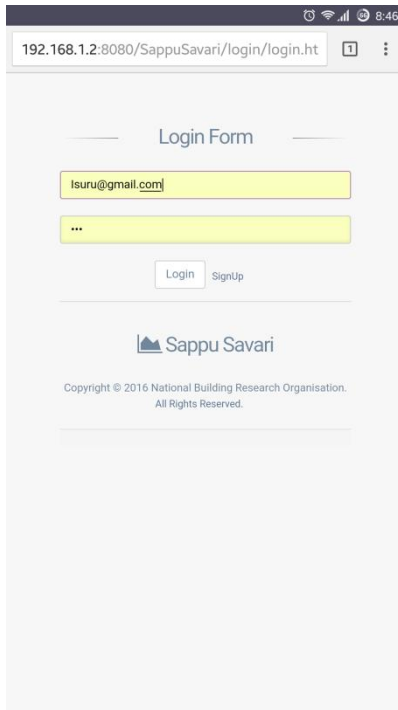


Figure 33: Login

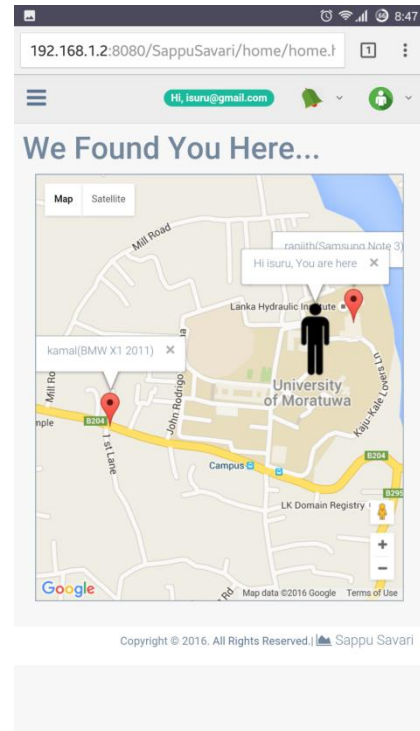


Figure 34: Home

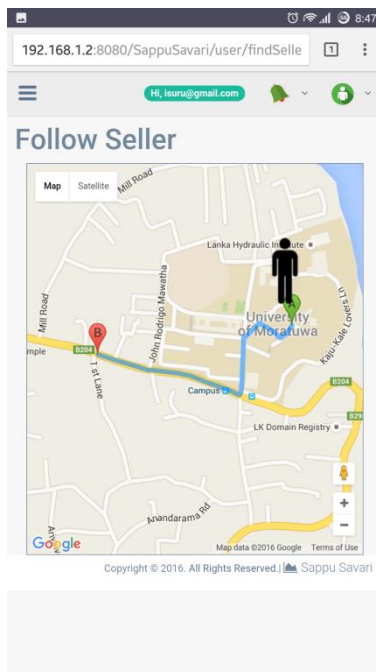


Figure 35: Follow Seller

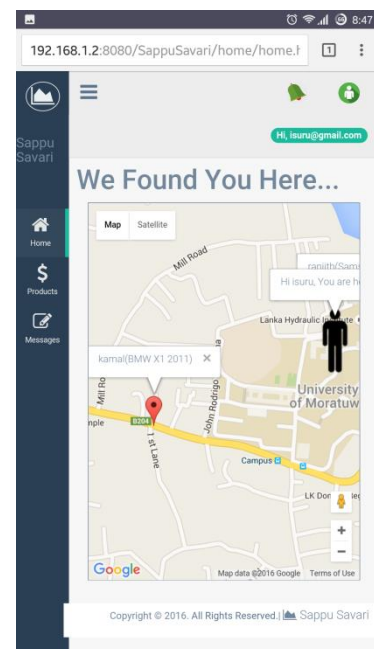


Figure 36: Home Menu

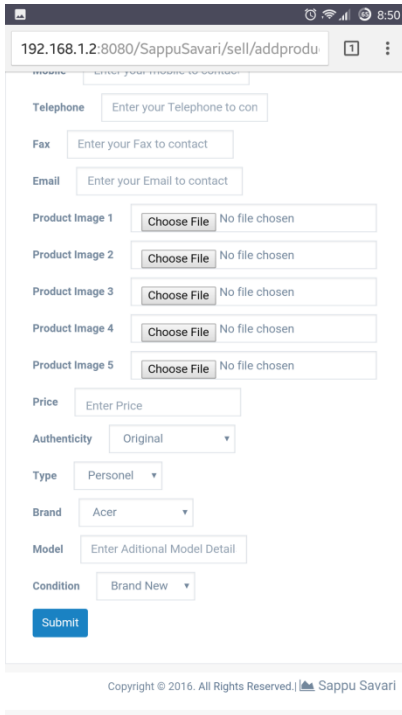


Figure 37: Add Product

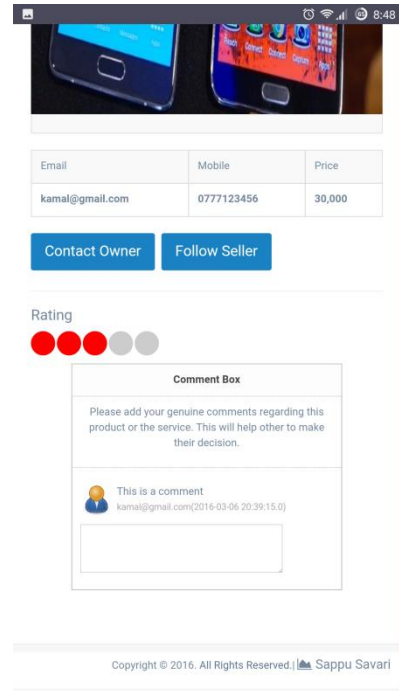


Figure 38: View Product

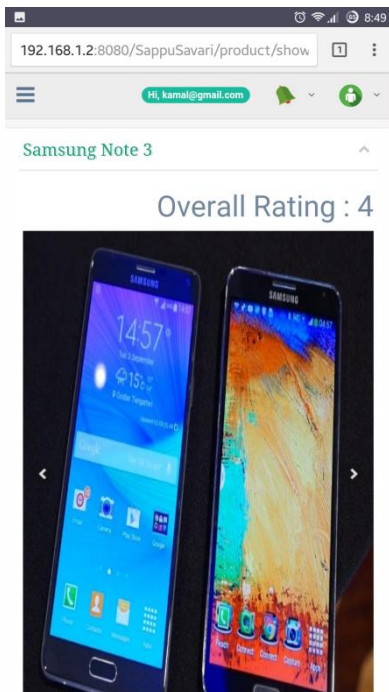


Figure 39: View Product Images

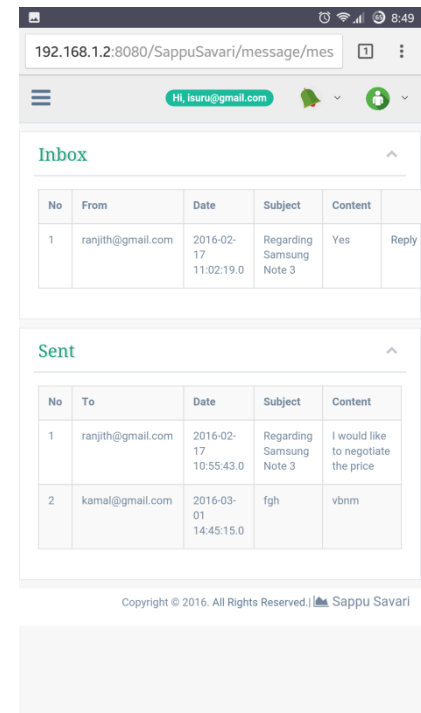


Figure 40: Message Dashboard

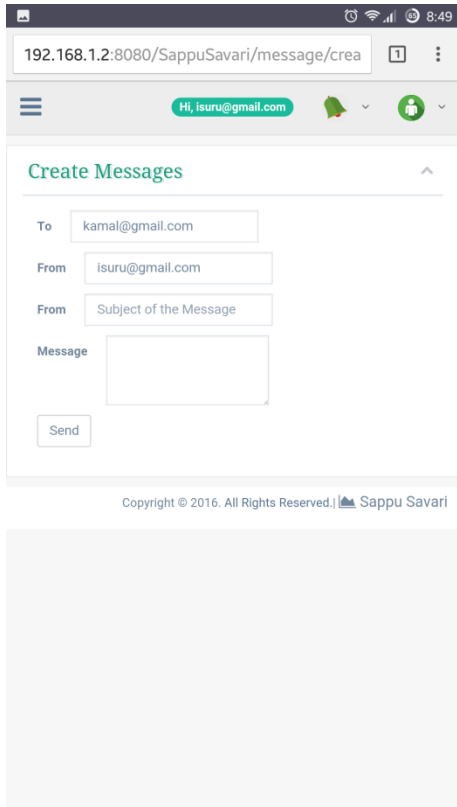


Figure 42: Create Messages

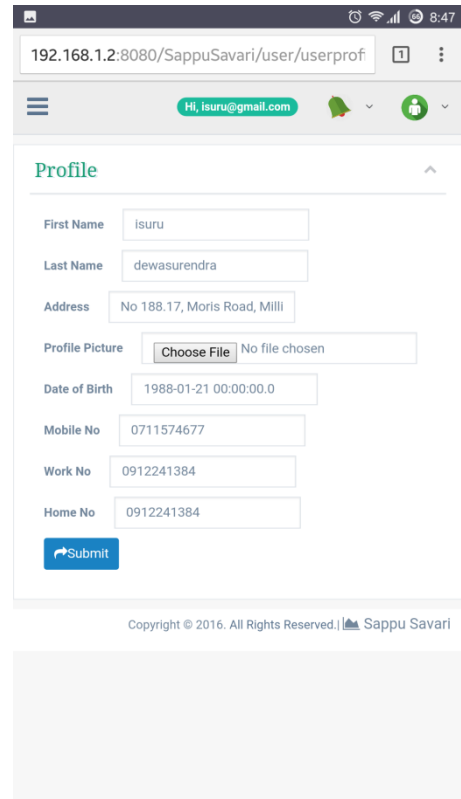


Figure 41: User Profile

9.5 Appendix E – Source Code

9.5.1 Back End Java Code

9.5.1.1 Product Domain

```
public class Product implements Serializable{

    private static final long serialVersionUID = 5489679679570043539L;
    private long productId;
    private String productTitle;
    private String description;
    private String address;
    private String mobile;
    private String landPhone;
    private String fax;
    private String email;
    private double price;
    private String productType;
    private String productBrand;
    private String productModel;
    private String authenticity;
    private String modelYear;
    private String productCondition;
    private String millege;
    private String transmission;
    private String fuelType;
    private String bodyType;
    private String productSize;
    private int noOfBathroom;
    private int noOfRoom;
    private String productStatus;
    private Date createdDateTime;
    private Date updatedDateTime;
    private SystemUser user;
    private PopularArea popularArea;
    private List<Favorite> favoriteList;
    private List<Comment> commentList;
    private List<Rating> ratingList;
    private ProductSubCategory productSubCategory;
    private List<ProductMultimedia> multiMediaList;

}
```

9.5.1.2 Product Category Domain

```
public class ProductCategory implements Serializable{

    private static final long serialVersionUID = 5489679679570043539L;

    private long productCategoryId;
    private String categoryName;
```

```
        private List<ProductSubCategory> productSubCategoryList;
    }
```

9.5.1.3 Product Sub Category Domain

```
public class ProductSubCategory implements Serializable{

    private static final long serialVersionUID = 5489679679570043539L;

    private long productSubCategoryId;
    private String subCategoryName;
    private List<Product> productList;
    private List<SearchRequest> searchRequestList;
    private ProductCategory productCategory;

}
```

9.5.1.4 Product Multimedia Domain

```
public class ProductMultimedia implements Serializable{

    private static final long serialVersionUID = 5489679679570043539L;
    private long productMultimediaId;
    private String multimediaType;
    private String media;
    private Date updatedDateTime;
    private Product product;

}
```

9.5.1.5 Product Search Request Domain

```
public class SearchRequest implements Serializable{

    private static final long serialVersionUID = 5489679679570043539L;
    private long searchRequestId;
    private String title;
    private String description;
    private String address;
    private double minPrice;
    private double maxPrice;
    private String authenticity;
    private String productType;
    private String brand;
    private String model;
    private String productCondition;
    private String modelYear;
    private int minMillege;
    private int maxMillege;
    private String transmission;
    private String fuelType;
    private String bodyType;
    private int minNoOfBathroom;
    private int maxNoOfBathroom;

}
```

```

    private int minNoOfRoom;
    private int maxNoOfRoom;
    private PopularArea popularArea;
    private ProductSubCategory productSubCategory;
    private SystemUser user;
    private Date createdDateTime;
}

```

9.5.1.6 Notification Domain

```

public class Notification implements Serializable{

    private static final long serialVersionUID = 5489679679570043539L;
    private long notificationId;
    private int seenStatus;
    private String notificationType;
    private String description;
    private int priority;
    private Date lastSeenTime;
    private Date updatedDateTime;
    private SystemUser user;
    private Product product;
}

```

9.5.1.7 Rating Domain

```

public class Rating implements Serializable{

    private static final long serialVersionUID = 5489679679570043539L;
    private long ratingId;
    private int rating;
    private Date updatedDateTime;
    private SystemUser user;
    private Product product;
}

```

9.5.1.8 Message Domain

```

public class Message implements Serializable{

    private static final long serialVersionUID = 5489679679570043539L;
    private long messageId;
    private String subject;
    private String message;
    private Date createdDateTime;
    private Date updatedDateTime;
    private String readStatus;
    private SystemUser sender;
}

```

```

        private SystemUser receiver;
        private long productId;
    }

```

9.5.1.9 Comment Domain

```

public class Comment implements Serializable{

    private static final long serialVersionUID = 5489679679570043539L;
    private long commentId;
    private String comment;
    private String showStatus;
    private Date updatedDateTime;
    private SystemUser user;
    private Product product;
}

```

9.5.1.10 User Domain

```

public class SystemUser implements Serializable{

    private static final long serialVersionUID = 5489679679570043539L;
    private long userId;
    private String username;
    private String password;
    private String firstName;
    private String lastName;
    private String address;
    private String profilePic;
    private Date dateOfBirth;
    private String mobileNo;
    private String workNo;
    private String homeNo;
    private double latitude;
    private double longitude;
    private int accuracy;
    private int activeStatus;
    private List<Favorite> favoriteList;
    private List<Comment> commentList;
    private List<Rating> ratingList;
    private List<Message> messageSentList;
    private List<Message> messageRecievedList;
    private List<Product> productList;
    private List<SystemUserRole> systemUserRoleList;
    private List<SearchRequest> searchRequestList
}

```

9.5.1.11 User Role Domain

```

public class UserRole implements GrantedAuthority {

```

```

private static final long serialVersionUID = 648258389880086882L;
private int userRoleId;
private List<SystemUserRole> systemUserRoles;
private String authority;
}

```

9.5.1.12 Product Service

```

public interface ProductService {
    public Product saveUpdateProduct(Product product);
    public Product getProductById(long productId);
    public List<Product> getAllProducts();
    public List<Product> getAllProductByUserId(long userId);
    public List<Product> getSearchedProductList(SearchRequest
searchRequest);
    public List<Product> getFavoriteProductsByUser(long userId);
    public List<Product> getCommentedProductsByUser(long userId);
}

```

9.5.1.13 Product Service Implementation

```

public class ProductServicesImpl implements ProductService{

    public List<Product> getAllProducts() {
        return productDao.findAll();
    }

    public List<Product> getAllProductByUserId(long userId) {
        SystemUser user = new SystemUser();
        user.setUserId(userId);
        return productDao.findProductByUser(user);
    }

    public Product saveUpdateProduct(Product product) {
        return productDao.save(product);
    }

    public List<Product> getSearchedProductList(SearchRequest searchRequest)
    {

        ProductSubCategory productSubCategory =
searchRequest.getProductSubCategory();
        ProductCategory productCategory =
productSubCategory.getProductCategory();

        List<Product> productList = new ArrayList<Product>();

```

```

}if(productCategory.getCategoryName().equals(EnumConstant.ProductCategory.PROPERTY.toString())){

    productList = productDao.getPropertyProductList(

        searchRequest.getProductSubCategory().getProductSubCategoryId(),

        (searchRequest.getMinPrice()==0)?Double.MIN_VALUE:searchRequest.
        getMinPrice(),

        (searchRequest.getMaxPrice()==0)?Double.MAX_VALUE:searchRequest.
        getMaxPrice(),
        searchRequest.getProductType(),
        (searchRequest.getProductType()==null)?1:0,

        searchRequest.getProductCondition(),
        (searchRequest.getProductCondition()==null)?1:0,

        (searchRequest.getMinNoOfBathroom()==0)?Integer.MIN_VALUE:
        searchRequest.getMinNoOfBathroom(),

        (searchRequest.getMaxNoOfBathroom()==0)?Integer.MAX_VALUE:
        searchRequest.getMaxNoOfBathroom(),

        (searchRequest.getMinNoOfRoom()==0)?Integer.MIN_VALUE:
        searchRequest.getMinNoOfRoom(),

        (searchRequest.getMaxNoOfRoom()==0)?Integer.MAX_VALUE:
        searchRequest.getMaxNoOfRoom()
    );
}

return productList;
}

public Product getProductById(long productId) {
    return productDao.findOne(productId);
}

public List<Product> getFavoriteProductsByUser(long userId) {
    try {
        List<Favorite> favoriteList =
            favoriteService.getFavoritesByUserId(userId);
        List<Product> productList = new ArrayList<Product>();
        for (Favorite favorite : favoriteList) {
            productList.add(productDao.findOne(
                favorite.getProduct().getProductId()));
        }
        return productList;
    } catch (Exception e) {
        System.out.println("error occured when get products for
            the user's favorite selection");
        e.printStackTrace();
        return null;
    }
}

```

```

    }

    public List<Product> getCommentedProductsByUser(long userId) {
        try {
            List<Comment> commentList =
                commentService.getCommentListByUserId(userId);
            List<Product> productList = new ArrayList<Product>();
            for (Comment comment : commentList) {

                productList.add(productDao.findOne(
                    comment.getProduct().getProductId()));
            }
            return productList;
        } catch (Exception e) {
            System.out.println("error occured when get products for
                the user's comments");
            e.printStackTrace();
            return null;
        }
    }
}

```

9.5.1.14 Product Category Service

```

import com.isd.sappu.savari.domains.ProductCategory;

public interface ProductCategoryService {

    public ProductCategory saveOrUpdateProductCategory(ProductCategory
productCategory);
    public List<ProductCategory> getAllProductCategory();
}

```

9.5.1.15 Product Category Service Implementation

```

@Service
public class ProductCategoryServiceImpl implements ProductCategoryService{
    public ProductCategory saveOrUpdateProductCategory(ProductCategory
productCategory) {
        return productCategoryDao.save(productCategory);
    }
    public List<ProductCategory> getAllProductCategory() {
        return productCategoryDao.findAll();
    }
}

```

9.5.1.16 Product Multimedia Service

```

import com.isd.sappu.savari.domains.ProductMultimedia;

```



```

public interface ProductMultimediaService {
    public List<ProductMultimedia> getMultimediaList(long productId);
}

```

9.5.1.17 Product Multimedia Service Implementation

```

public class ProductMultimediaServiceImpl implements ProductMultimediaService{
    public List<ProductMultimedia> getMultimediaList(long productId) {
        Product product = new Product();
        product.setProductId(productId);
        return productMultimediaDao.findByProduct(product);
    }
}

```

9.5.1.18 Location Service

```

public interface LocationService {

    public double getDistance(double fromLat, double fromLon, double toLat,
double toLon, char unit);
}

```

9.5.1.19 Location Service Implementation

```

public class LocationServiceImpl implements LocationService {
    public double getDistance(double fromLat, double fromLon, double toLat,
double toLon, char unit) {
        double theta = fromLon - toLon;
        double dist = Math.sin(deg2rad(fromLat)) *
Math.sin(deg2rad(toLat)) + Math.cos(deg2rad(fromLat)) *
Math.cos(deg2rad(toLat)) * Math.cos(deg2rad(theta));
        dist = Math.acos(dist);
        dist = rad2deg(dist);
        dist = dist * 60 * 1.1515;
        if (unit == 'K') {
            dist = dist * 1.609344;
        } else if (unit == 'N') {
            dist = dist * 0.8684;
        }
        return (dist);
    }

    private double deg2rad(double deg) {
        return (deg * Math.PI / 180.0);
    }

    private double rad2deg(double rad) {
        return (rad * 180 / Math.PI);
    }
}

```

```
    }  
}
```

9.5.1.20 Search Request Service

```
public interface SearchRequestService {  
    public SearchRequest getSearchRequestById(long searchRequestId);  
    public SearchRequest saveUpdateSearchRequest(SearchRequest  
searchRequest);  
    public void deleteSearchRequest(long searchRequestId);  
    public String deleteUserSearchRequest(long userId);  
    public List<SearchRequest> getAllSearchRequestByUserId(long userId);  
}
```

9.5.1.21 Search Request Service Implementation

```
public class SearchRequestServiceImpl implements SearchRequestService{  
  
    public SearchRequest getSearchRequestById(long searchRequestId) {  
        return searchRequestDao.findOne(searchRequestId);  
    }  
  
    public SearchRequest saveUpdateSearchRequest(SearchRequest  
searchRequest) {  
        return searchRequestDao.save(searchRequest);  
    }  
  
    public List<SearchRequest> getAllSearchRequestByUserId(long userId) {  
        SystemUser user = new SystemUser();  
        user.setUserId(userId);  
        return searchRequestDao.findByUser(user);  
    }  
  
    public void deleteSearchRequest(long searchRequestId) {  
        searchRequestDao.delete(searchRequestId);  
    }  
  
    public String deleteUserSearchRequest(long userId) {  
        List<SearchRequest> searchRequestList =  
this.getAllSearchRequestByUserId(userId);  
        for (SearchRequest searchRequest : searchRequestList) {  
  
this.deleteSearchRequest(searchRequest.getSearchRequestId());  
        }  
        return EnumConstant.ReturnStatus.SUCCESS.toString();  
    }  
}
```

9.5.1.22 Rating Service

```
public interface RatingService {
    public Rating saveUpdateRating(Rating rating);
    public Rating getRating(long ratingId);
    public Rating getRating(long userId, long productId);
    public List<Rating> getRatingsByUserId(long userId);
    public List<Rating> getRatingsByProductId(long productId);
    public void deleteRating(Rating rating);
    public int calculateOverallRating(long productId);
}
```

9.5.1.23 Rating Service Implementation

```
public class RatingServiceImpl implements RatingService {

    public Rating saveUpdateRating(Rating rating) {
        Rating rate = this.getRating(rating.getUser().getUserId(),
rating.getProduct().getProductId());
        if(rate != null){
            rate.setRating(rating.getRating());
            return ratingDao.save(rate);
        }else{
            return ratingDao.save(rating);
        }
    }

    public Rating getRating(long ratingId) {
        return ratingDao.findByRatingId(ratingId);
    }

    public List<Rating> getRatingsByUserId(long userId) {
        SystemUser user = new SystemUser();
        user.setUserId(userId);
        return ratingDao.findByUser(user);
    }

    public List<Rating> getRatingsByProductId(long productId) {
        Product product = new Product();
        product.setProductId(productId);
        return ratingDao.findByProduct(product);
    }

    public void deleteRating(Rating rating) {
        ratingDao.delete(rating);
    }

    public int calculateOverallRating(long productId) {
        List<Rating> ratingList = this.getRatingsByProductId(productId);
        int ratingSum = 0;
        int ratingCount = 0;

        for (Rating rating : ratingList) {
```

```

        ratingCount++;
        ratingSum = ratingSum + rating.getRating();
    }

    if(ratingCount>0 && ratingSum>0){
        return ratingSum/ratingCount;
    }else{
        return 0;
    }
}

public Rating getRating(long userId, long productId) {
    SystemUser user = new SystemUser();
    user.setUserId(userId);
    Product product = new Product();
    product.setProductId(productId);
    return ratingDao.findByUserAndProduct(user, product);
}
}
}

```

9.5.1.24 Comment Service

```

public interface CommentService {
    public Comment saveUpdateComment(Comment comment);
    public Comment getComment(long commentId);
    public void deleteComment(Comment comment);
    public List<Comment> getCommentListByProductId(long productId);
    public List<Comment> getCommentListByUserId(long userId);
}

```

9.5.1.25 Comment Service Implementation

```

public class CommentServiceImpl implements CommentService{

    public Comment saveUpdateComment(Comment comment) {
        return commentDao.save(comment);
    }

    public Comment getComment(long commentId) {
        return commentDao.findByCommentId(commentId);
    }

    public void deleteComment(Comment comment) {
        commentDao.delete(comment);
    }

    public List<Comment> getCommentListByProductId(long productId) {
        Product product = new Product();
        product.setProductId(productId);
        return commentDao.findCommentByProduct(product);
    }
}

```

```

    }

    public List<Comment> getCommentListByUserId(long userId) {
        SystemUser systemUser = new SystemUser();
        systemUser.setUserId(userId);
        return commentDao.findCommentByUser(systemUser);
    }
}

```

9.5.1.26 Message Service

```

public interface MessageService {

    public Message saveUpdateMessage(Message message);
    public Message getMessage(long messageId);
    public List<Message> getMessagesBySender(long userId);
    public List<Message> getMessagesByReceiver(long userId);
    public void deleteMessage(Message message);

}

```

9.5.1.27 Message Service Implementation

```

public class MessageServiceImpl implements MessageService {

    public Message saveUpdateMessage(Message message) {
        return messageDao.save(message);
    }
    public Message getMessage(long messageId) {
        return messageDao.findOne(messageId);
    }
    public List<Message> getMessagesBySender(long userId) {
        SystemUser sender = new SystemUser();
        sender.setUserId(userId);
        return messageDao.findBySender(sender);
    }
    public List<Message> getMessagesByReceiver(long userId) {
        SystemUser reciever = new SystemUser();
        reciever.setUserId(userId);
        return messageDao.findByReceiver(reciever);
    }
    public void deleteMessage(Message message) {
        messageDao.delete(message);
    }
}

```

9.5.1.28 User Service

```

public interface SystemUserService {

```

```

    public SystemUser saveUpdateSystemUser(SystemUser user);
    public SystemUser getSystemUserByUsername(String username);
    public SystemUser getSystemUser(long userId);
    public void deleteSystemUser(SystemUser user);
    public List<SystemUser> getAllSystemUsers();
}

```

9.5.1.29 User Service Implementation

```

public class SystemUserServiceImpl implements SystemUserService{
    public SystemUser saveUpdateSystemUser(SystemUser user) {
        return systemUserDao.save(user);
    }
    public SystemUser getSystemUser(long userId) {
        return systemUserDao.findOne(userId);
    }

    public void deleteSystemUser(SystemUser user) {
        systemUserDao.delete(user);
    }
    public SystemUser getSystemUserByUsername(String username) {
        return systemUserDao.findByUsername(username);
    }
    public List<SystemUser> getAllSystemUsers() {
        return systemUserDao.findAll();
    }
}

```

9.5.1.30 Scheduler Service Implementation

```

public class ScheduleLocationSearchService {

    @Scheduled(fixedDelay = 1000 * 20)
    public void demoServiceMethod() {
        List<SystemUser> applicationUserList =
            systemUserService.getAllSystemUsers();

        for (SystemUser applicationUser : applicationUserList) {

            List<Product> relevantProducts = new ArrayList<Product>();
            List<Product> searchedProductList = new
                ArrayList<Product>();

            List<Product> favoriteProducts =
                productService.getFavoriteProductsByUser(applicationUser.g
                    etUserId());
            relevantProducts.addAll(favoriteProducts);
        }
    }
}

```

```

List<Product> commentedProducts =
productService.getCommentedProductsByUser(applicationUser.
getUserId());
relevantProducts.addAll(commentedProducts);

List<SearchRequest> searchRequests = searchRequestService
.getAllSearchRequestByUserId(
applicationUser.getUserId());
for (SearchRequest searchRequest : searchRequests) {
List<Product> searchedProducts =
productService.getSearchedProductList(searchRequest)
;
relevantProducts.addAll(searchedProducts);
searchedProductList.addAll(searchedProducts);
}

for (Product product : relevantProducts) {
double fromLat = applicationUser.getLatitude();
double fromLon = applicationUser.getLongitude();
double toLat = product.getUser().getLatitude();
double toLon = product.getUser().getLongitude();
double distance =
this.getDistanceFromLatLonInKm(fromLat, fromLon, toLat,
toLon);
if (distance < 1) {
Notification notification = new
Notification(0,
EnumConstant.NotificationType.DISTANCE.toStri
ng(), "your seller is near by", 5, null, new
Date(), applicationUser, product);
List<Notification> existNotifications =
notificationService.getNotification(applicati
onUser.getUserId(), product.getProductId(),
EnumConstant.NotificationType.DISTANCE.toStri
ng());
if(existNotifications == null ||
(existNotifications != null &&
existNotifications.size() == 0)){

notificationService.
saveUpdateNotification(notification);
}
}

for (Product product : searchedProductList) {
EnumConstant.NotificationType.SEARCH.toString());
if(existNotifications == null || (existNotifications
!= null && existNotifications.size() == 0)){

notificationService.
saveUpdateNotification(notification);
}
}
}

```

```

    }
}

public double getDistanceFromLatLonInKm(double fromLat, double
fromLon, double toLat, double toLon) {
try {
    int R = 6371; // Radius of the earth in km
    double dLat = deg2rad(toLat - fromLat); // deg2rad below
    double dLon = deg2rad(toLon - fromLon);
    double a = Math.sin(dLat / 2) * Math.sin(dLat / 2)
        + Math.cos(deg2rad(fromLat)) *
        Math.cos(deg2rad(toLat)) * Math.sin(dLon / 2) *
        Math.sin(dLon / 2);
    double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
    double d = R * c; // Distance in km
    return d;
} catch (Exception e) {
    e.printStackTrace();
    return -1;
}
}

public double deg2rad(double deg) {
    return deg * (Math.PI / 180);
}
}
}

```

9.5.1.31 Product Data Access Service

```

public interface ProductDao extends CrudRepository<Product, Long>{

    public Product findById(long productId);
    public List<Product> findAll();
    public List<Product> findProductByUser(SystemUser user);

    public List<Product> getOtherProductList(
        @Param("productSubCategoryId") long productCategoryId,
        @Param("minPrice") double minPrice,
        @Param("maxPrice") double maxPrice,
        @Param("authenticity") String authenticity,
        @Param("authenticityBoolVal") int authenticityBoolVal,
        @Param("productType") String productType,
        @Param("productTypeBoolVal") int productTypeBoolVal,
        @Param("productBrand") String productBrand,
        @Param("productBrandBoolVal") int productBrandBoolVal,
        @Param("productCondition") String productCondition,
        @Param("productConditionBoolVal") int
        productConditionBoolVal);

    public List<Product> getVehicleProductList(

```



```

        @Param("productSubCategoryId") long productCategoryId,
        @Param("minPrice") double minPrice,
        @Param("maxPrice") double maxPrice,
        @Param("productType") String productType,
        @Param("productTypeBoolVal") int productTypeBoolVal,
        @Param("productCondition") String productCondition,
        @Param("productConditionBoolVal") int
        productConditionBoolVal,
        @Param("minMillege") int minMillege,
        @Param("maxMillege") int maxMillege,
        @Param("transmission") String transmission,
        @Param("transmissionBoolVal") int transmissionBoolVal,
        @Param("fuelType") String fuelType,
        @Param("fuelTypeBoolVal") int fuelTypeBoolVal,
        @Param("bodyType") String bodyType,
        @Param("bodyTypeBoolVal") int bodyTypeBoolVal
    );

    public List<Product> getPropertyProductList(
        @Param("productSubCategoryId") long productCategoryId,
        @Param("minPrice") double minPrice,
        @Param("maxPrice") double maxPrice,
        @Param("productType") String productType,
        @Param("productTypeBoolVal") int productTypeBoolVal,
        @Param("productCondition") String productCondition,
        @Param("productConditionBoolVal") int
        productConditionBoolVal,
        @Param("minNoOfBathroom") int minNoOfBathroom,
        @Param("maxNoOfBathroom") int maxNoOfBathroom,
        @Param("minNoOfRoom") int minNoOfRoom,
        @Param("maxNoOfRoom") int maxNoOfRoom
    );
}

```

9.5.1.32 Product Category Data Access Service

```

public interface ProductCategoryDao extends CrudRepository<ProductCategory,
Long>{
    public List<ProductCategory> findAll();
}

```

9.5.1.33 Product Sub Category Data Access Service

```

public interface ProductSubCategoryDao extends
CrudRepository<ProductSubCategory, Long>{
    public ProductSubCategory findByProductSubCategoryId(long
productSubCategoryId);
    public List<ProductSubCategory> findAll();
}

```

```
        public List<ProductSubCategory> findByProductCategory(ProductCategory
productCategory);
    }
```

9.5.1.34 Product Multimedia Data Access Service

```
public interface ProductMultimediaDao extends
CrudRepository<ProductMultimedia, Long>{

    public List<ProductMultimedia> findByProduct(Product product);

}
```

9.5.1.35 Search Request Data Access Service

```
public interface SearchRequestDao extends CrudRepository<SearchRequest, Long>{
    public SearchRequest findById(long searchRequestId);
    public List<SearchRequest> findByUser(SystemUser user);
}
}
```

9.5.1.36 Rating Data Access Service

```
public interface RatingDao extends CrudRepository<Rating, Long>{
    public Rating findById(long ratingId);
    public Rating findByUserAndProduct(SystemUser user, Product product);
    public List<Rating> findByUser(SystemUser user);
    public List<Rating> findByProduct(Product product);
}
}
```

9.5.1.37 Comment Data Access Service

```
public interface CommentDao extends CrudRepository<Comment, Long>{
    public Comment findById(long commentId);
    public List<Comment> findCommentByProduct(Product product);
    public List<Comment> findCommentByUser(SystemUser systemUser);
}
}
```

9.5.1.38 Message Data Access Service

```
public interface MessageDao extends CrudRepository<Message, Long>{
    public Message findById(long messageId);
    public List<Message> findBySender(SystemUser sender);
    public List<Message> findByReceiver(SystemUser receiver);
}
}
```

9.5.1.39 User Data Access Service

```
public interface SystemUserDao extends CrudRepository<SystemUser, Long>{
    public SystemUser findByUsername(String username);
    public SystemUser findByUserId(long userId);
    public List<SystemUser> findAll();
}
```

9.5.1.40 Login Success Service

```
public class LoginSuccessHandler implements AuthenticationSuccessHandler {

    private AuthenticationSuccessHandler target = new
    SavedRequestAwareAuthenticationSuccessHandler();

    public void onAuthenticationSuccess(HttpServletRequest request,
        HttpServletResponse response, Authentication
        authentication) throws IOException,
        ServletException {
        System.out.println("CAME HERE onAuthenticationSuccess");

        if(authentication != null){
            Object principal = authentication.getPrincipal();
            UserDetails userDetails = (UserDetails) (principal
            instanceof UserDetails ? principal : null);

            SystemUser user =
            systemUserService.getSystemUserByUsername(userDetails.getU
            sername());
            sessionUtil.addLoggedUserToSession(AppConstant.LOGGED_USER
            , user, request);

            if(userDetails != null){
                try {
                    target.onAuthenticationSuccess(request,
                    response, authentication);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

9.5.1.41 Login Failure Service

```
package com.isd.sappu.savari.services;

@Component(value="loginFailureHandler")
public class LoginFailureHandler implements AuthenticationFailureHandler {

    @Override
    public void onAuthenticationFailure(HttpServletRequest request,
```

```

        HttpServletResponse response, AuthenticationException
        arg2)
        throws IOException, ServletException {

        response.sendRedirect("login/login.htm?login_error=t");

    }
}

```

9.5.2 Front End HTML/JAVASCRIPT Code

9.5.2.1 Home Page

```

$("document").ready(function(){
    var firstName = "";

    $.get("../user/getSystemUser.do",{
        userId:${sessionScope.LOGGED_USER.userId}
    },function(data, status){
        var fromLatitude = data.latitude;
        var fromLongitude = data.longitude;
        console.log(fromLatitude);
        console.log(fromLongitude);
        firstName = data.firstName;
        initializer(fromLatitude, fromLongitude);
    });

    function initializer(latitude, longitude){
        var myCenter=new google.maps.LatLng(latitude, longitude);

        var map = new google.maps.Map(document.getElementById('map-
        canvas'), {
            zoom: 16,
            center: myCenter
        });

        var marker=new google.maps.Marker({
            position:myCenter,
            animation:google.maps.Animation.BOUNCE,
            url:"/hhfhf.htm",
            icon: '../images/man_icon.png'
        });
        marker.setMap(map);
        var infowindow = new google.maps.InfoWindow({
            content:"Hi "+firstName+", You are here"

        });
        infowindow.open(map,marker);

        <c:forEach var="followSeller" items="${followSellers}"
        varStatus="count">

```

```

        var myCenter_{{count.index}}=new
        google.maps.LatLng('{{followSeller.sellerLatitude}}',
        '{{followSeller.sellerLongitude}}');
        var marker_{{count.index}}=new google.maps.Marker({
            position:myCenter_{{count.index}},
            animation:google.maps.Animation.DROP
        });
        marker_{{count.index}}.addListener('click', function() {

        window.location="../user/findSeller.htm?sUserId={{followSeller.sellerId}}";
        });
        marker_{{count.index}}.setMap(map);
        var infowindow_{{count.index}} = new
        google.maps.InfoWindow({

        content:"{{followSeller.sellerName}}
        ({{followSeller.productTitle})"
        });
        infowindow_{{count.index}}.open(map,marker_{{count.index}});
    </c:forEach>
    }
});
</script>

```

9.5.2.2 Follow Seller

```

<script type="text/javascript">
$("document").ready(function(){

    var fromlat = {{buyer.latitude}};
    var fromlong = {{buyer.longitude}};

    window.setInterval(function(){
        $.get("../user/getSystemUser.do",{
            userId:{{sessionScope.LOGGED_USER.userId}}
        },function(data, status){
            fromlat = data.latitude;
            fromlong = data.longitude;
            console.log(fromlat);
            console.log(fromlong);
        });
    }, 1000);

    var Center = new google.maps.LatLng({{buyer.latitude}},
    {{buyer.longitude}});
    var directionsDisplay;
    var directionsService = new google.maps.DirectionsService();
    var map;

    function initialize() {

```

```

directionsDisplay = new google.maps.DirectionsRenderer();
var properties = {
  center: Center,
  zoom: 20,
  mapTypeId: google.maps.MapTypeId.MAP
};

map = new google.maps.Map(document.getElementById("map"), properties);
directionsDisplay.setMap(map);

var marker = new google.maps.Marker({
  position: Center,
  animation: google.maps.Animation.BOUNCE,
  icon: '../images/man_icon.png'
});

window.setInterval(function(){
  changeMarkerPosition(marker);
}, 1000);

marker.setMap(map);
Route();
}

function changeMarkerPosition(marker) {
  var latlng = new google.maps.LatLng(fromlat, fromlong);
  marker.setPosition(latlng);
}

function Route() {

  var start = new google.maps.LatLng(${buyer.latitude},
${buyer.longtitude});
  var end = new google.maps.LatLng(${seller.latitude},
${seller.longtitude});
  var request = {
    origin: start,
    destination: end,
    travelMode: google.maps.TravelMode.DRIVING
  };
  directionsService.route(request, function(result, status) {
    if (status == google.maps.DirectionsStatus.OK) {
      directionsDisplay.setDirections(result);
    } else {
      alert("couldn't get directions:" + status);
    }
  });
}

google.maps.event.addDomListener(window, 'load', initialize);
});
</script>

```

9.5.2.3 User Location Identification

```
function success(pos) {
    var crd = pos.coords;
    $.post("../user/updateUserLocation.do",
        {
            userId:"${sessionScope.LOGGED_USER.userId}",
            latt: crd.latitude,
            longt: crd.longitude,
            acc:crd.accuracy
        },
        function(data, status){
            console.log("location saved...");
        });
};

function error(err) {
    console.warn('ERROR(' + err.code + '): ' + err.message);
};

var options = {
    enableHighAccuracy: true,
    timeout: 5000,
    maximumAge: 0
};

navigator.geolocation.getCurrentPosition(success, error, options);
```

9.6 Appendix F – Evaluation and Testing

Sappu Savari – Location Based Advertising and Marketing User Evaluation Criteria for the System

General Instructions:

You may circle the appropriate number appear in cages to represent your choice.

1	2	3	4	5
Outstanding	Good	Satisfactory	Poor	Unsatisfactory

Skip question/s if not applicable.

User Information

- a. Select your role

1. General user with IT knowledge	2. QA	3. IT Expert	4. Other
-----------------------------------	-------	--------------	----------

- b. What is your age group?

1. 18-23 years	2. 24-29 years	3. 30-35 years	4. 35-55 years	5. More than 55
----------------	----------------	----------------	----------------	-----------------

- c. Gender

1. Male	2. Female
---------	-----------

The information provided in the application

- j. Content is free from spelling and grammatical errors.

1	2	3	4	5
---	---	---	---	---

- k. The information provided on the site is clear, concise and informative to the intended user.

1	2	3	4	5
---	---	---	---	---

- l. Information given in the site is helpful for a learner.

1	2	3	4	5
---	---	---	---	---

- m. The language is non-discriminatory. Content is free from race, ethnic, gender, age and other stereotype.

1	2	3	4	5
---	---	---	---	---

- n. By using the information given in the site, consumers are able to solve most of their problems.

1	2	3	4	5
---	---	---	---	---

- o. All the needed information is available in the system, as an example the mobile phone brand list.

1	2	3	4	5
---	---	---	---	---

- p. Information given in non-technical, understandable language without technical jargons.

1	2	3	4	5
---	---	---	---	---

- q. The information is well structured in the application, suitable for marketing purposes.

1	2	3	4	5
---	---	---	---	---

- r. Information given in the system not confusing the operator

1	2	3	4	5
---	---	---	---	---

User satisfaction on interfaces

- j. Reading of letters of application

1	2	3	4	5
---	---	---	---	---

- k. Organization of information.

1	2	3	4	5
---	---	---	---	---

- l. Use of terminology throughout the application.

1	2	3	4	5
---	---	---	---	---

- m. Proper navigation.

1	2	3	4	5
---	---	---	---	---

- n. Design of a page provides relevant information only.

1	2	3	4	5
---	---	---	---	---

- o. Audit Information is visible in the application.

1	2	3	4	5
---	---	---	---	---

- p. Position of messages on the screen is easy to view.

1	2	3	4	5
---	---	---	---	---

- q. Color choices visually accessible and pleasant to see.

1	2	3	4	5
---	---	---	---	---

- r. The site achieves its purpose.

1	2	3	4	5
---	---	---	---	---

Usability

- j. The layout and the design of the application encourage frequent use of the application.

1	2	3	4	5
---	---	---	---	---

k. Easy navigation.

1	2	3	4	5
---	---	---	---	---

l. The organization is clear, logical and effective, making it easy and simple for the intended audience to understand.

1	2	3	4	5
---	---	---	---	---

m. Individuals can easily start and exit the program.

1	2	3	4	5
---	---	---	---	---

n. The individual has the choice of going directly to desire information or using a structure search to identify relevant topics.

1	2	3	4	5
---	---	---	---	---

o. Feel very confident when using the application.

1	2	3	4	5
---	---	---	---	---

p. Not necessary to get to know about the site before it could effectively use it.

1	2	3	4	5
---	---	---	---	---

s. Use the language of the application is clear to the intended audience.

1	2	3	4	5
---	---	---	---	---

t. Application can be used without written instructions.

1	2	3	4	5
---	---	---	---	---

Ease of Learning

g. Can learn to navigate the application quickly.

1	2	3	4	5
---	---	---	---	---

h. Can easily remember how to use the application next time.

1	2	3	4	5
---	---	---	---	---

i. It is easy to remember main and top many item names and use of commands.

1	2	3	4	5
---	---	---	---	---

j. Tasks can be performed in a straightforward manner.

1	2	3	4	5
---	---	---	---	---

k. Help messages on the screen.

1	2	3	4	5
---	---	---	---	---

l. Links to supplementary reference materials.

1	2	3	4	5
---	---	---	---	---

Overall Impression

j. User friendliness.

1	2	3	4	5
---	---	---	---	---

k. Make life easy with its facilities.

1	2	3	4	5
---	---	---	---	---

l. Usefulness.

- m. Ease of use.

1	2	3	4	5
---	---	---	---	---
- n. Efficiency

1	2	3	4	5
---	---	---	---	---
- o. The system is easy to learn and the information provided is meaningful and helpful.

1	2	3	4	5
---	---	---	---	---
- p. The application is a one-stop advertising system.

1	2	3	4	5
---	---	---	---	---
- q. Time taken for search products is acceptable.

1	2	3	4	5
---	---	---	---	---
- r. Time taken to find sellers are acceptable.

1	2	3	4	5
---	---	---	---	---

Evaluators Observation

- Information provided in the application

Evaluator	Age	Gender	a	b	c	d	e	f	g	h	i
General user with IT knowledge	30-35 year	male	2	2	1	1	2	1	1	2	2
General user with IT knowledge	24-29 year	male	2	3	2	2	2	1	2	2	3
General user with IT knowledge	30-35 year	male	1	1	2	1	2	2	1	1	1
General user with IT knowledge	24-29 year	female	1	2	2	1	2	2	1	1	1
General user with IT knowledge	24-29 year	female	1	2	3	1	3	3	2	3	2
General user with IT knowledge	24-29 year	female	1	1	2	1	2	2	1	2	2
General user with IT knowledge	30-35 year	female	1	3	3	1	2	3	2	2	3
General user with IT knowledge	30-35 year	male	1	2	2	1	1	2	1	1	1
General user with IT knowledge	24-29 year	male	1	1	2	1	3	2	1	2	1
General user with IT knowledge	24-29 year	male	1	2	2	1	2	2	1	1	1
QA	24-29 year	female	1	1	1	1	1	1	1	2	1
QA	30-35 year	female	2	1	2	1	1	2	1	2	2
IT Expert	24-29 year	female	1	1	2	1	1	1	1	1	2
IT Expert	24-29 year	male	1	1	3	2	2	1	2	1	2
IT Expert	30-35 year	female	1	1	2	1	1	1	1	1	2
IT Expert	24-29 year	female	1	2	2	1	1	1	2	1	1
IT Expert	24-29 year	female	2	2	1	1	1	2	1	2	2
Total			21	28	34	19	29	29	22	27	29
Average			1.2	1.6	2.2	1.1	1.7	1.7	1.3	1.6	1.7

Figure 43: Observation Summary – Information provided in the application

- **User satisfaction on Interfaces**

Evaluator	Age	Gender	a	b	c	d	e	f	g	h	i
General user with IT knowledge	30-35 year	male	2	1	2	1	1	2	3	2	1
General user with IT knowledge	24-29 year	male	2	1	2	3	1	2	1	1	2
General user with IT knowledge	30-35 year	male	2	2	2	1	1	2	1	1	1
General user with IT knowledge	24-29 year	female	1	3	2	1	1	1	2	1	2
General user with IT knowledge	24-29 year	female	1	2	2	1	2	2	2	1	2
General user with IT knowledge	24-29 year	female	1	2	2	1	1	2	2	1	1
General user with IT knowledge	30-35 year	female	1	3	2	2	2	2	2	1	2
General user with IT knowledge	30-35 year	male	1	2	2	1	1	1	1	1	1
General user with IT knowledge	24-29 year	male	1	2	1	1	1	1	1	1	1
General user with IT knowledge	24-29 year	male	1	1	2	1	2	2	2	1	1
QA	24-29 year	female	1	2	2	1	1	1	1	2	2
QA	30-35 year	female	1	1	1	2	1	2	2	1	1
IT Expert	24-29 year	female	1	1	2	1	1	2	2	1	1
IT Expert	24-29 year	male	1	1	2	1	2	1	2	1	1
IT Expert	30-35 year	female	1	2	3	2	1	2	1	2	2
IT Expert	24-29 year	female	1	2	1	2	1	1	1	2	1
IT Expert	24-29 year	female	1	1	2	1	1	2	2	3	1
Total			20	29	32	23	21	28	28	23	23
Average			1.2	1.7	1.9	1.4	1.2	1.6	1.6	1.4	1.4

Figure 44: Observation Summary – User satisfaction on interfaces

- Usability

Evaluator	Age	Gender	a	b	c	d	e	f	g	h	i
General user with IT knowledge	30-35 year	male	2	3	2	1	2	2	2	2	1
General user with IT knowledge	24-29 year	male	2	2	2	1	2	2	3	2	2
General user with IT knowledge	30-35 year	male	2	1	2	1	2	1	2	1	1
General user with IT knowledge	24-29 year	female	1	1	2	1	3	2	2	2	2
General user with IT knowledge	24-29 year	female	2	1	3	2	2	2	3	2	2
General user with IT knowledge	24-29 year	female	1	1	2	1	2	2	2	1	2
General user with IT knowledge	30-35 year	female	2	2	3	3	2	3	2	2	3
General user with IT knowledge	30-35 year	male	1	1	2	1	2	1	2	1	1
General user with IT knowledge	24-29 year	male	1	1	2	1	1	1	2	1	1
General user with IT knowledge	24-29 year	male	1	1	2	1	1	1	2	1	1
QA	24-29 year	female	1	1	2	1	2	1	1	1	2
QA	30-35 year	female	1	1	2	1	2	1	1	1	2
IT Expert	24-29 year	female	1	1	2	1	1	1	1	1	1
IT Expert	24-29 year	male	1	2	2	1	1	2	1	2	2
IT Expert	30-35 year	female	1	1	2	1	1	2	2	2	1
IT Expert	24-29 year	female	1	3	1	2	3	2	2	1	2
IT Expert	24-29 year	female	2	1	2	2	1	1	1	1	2
Total			23	24	35	22	30	27	31	24	28
Average			1.4	1.4	2.1	1.3	1.8	1.6	1.8	1.4	1.6

Figure 45: Observation Summary – Usability

- **Ease of Learning**

Evaluator	Age	Gender	a	b	c	d	e	f
General user with IT knowledge	30-35 year	male	2	1	2	2	2	5
General user with IT knowledge	24-29 year	male	2	2	2	2	2	4
General user with IT knowledge	30-35 year	male	1	1	1	2	1	2
General user with IT knowledge	24-29 year	female	1	1	2	1	3	5
General user with IT knowledge	24-29 year	female	2	1	3	2	2	2
General user with IT knowledge	24-29 year	female	1	1	1	2	1	2
General user with IT knowledge	30-35 year	female	2	2	2	1	2	3
General user with IT knowledge	30-35 year	male	1	1	2	1	1	4
General user with IT knowledge	24-29 year	male	2	1	1	1	1	2
General user with IT knowledge	24-29 year	male	1	1	2	1	2	3
QA	24-29 year	female	2	1	2	1	3	5
QA	30-35 year	female	1	2	1	1	1	5
IT Expert	24-29 year	female	1	1	1	1	2	3
IT Expert	24-29 year	male	1	1	2	1	1	2
IT Expert	30-35 year	female	2	2	1	3	2	2
IT Expert	24-29 year	female	3	2	2	1	1	2
IT Expert	24-29 year	female	1	2	2	1	3	5
Total			26	23	29	24	30	56
Average			1.5	1.4	1.7	1.4	1.8	3.3

Figure 46: Observation Summary – Ease of learning

- Overall Impression

Evaluator	Age	Gender	a	b	c	d	e	f	g	h	i
General user with IT knowledge	30-35 year	male	1	2	2	2	2	2	1	2	1
General user with IT knowledge	24-29 year	male	2	2	2	2	2	2	2	2	3
General user with IT knowledge	30-35 year	male	1	1	2	2	2	1	1	2	1
General user with IT knowledge	24-29 year	female	2	2	2	2	2	1	2	3	2
General user with IT knowledge	24-29 year	female	2	2	3	2	2	2	2	3	3
General user with IT knowledge	24-29 year	female	1	1	2	1	2	2	3	2	2
General user with IT knowledge	30-35 year	female	2	2	1	2	2	1	2	2	2
General user with IT knowledge	30-35 year	male	1	2	1	1	1	1	2	1	1
General user with IT knowledge	24-29 year	male	1	1	1	2	2	1	1	2	2
General user with IT knowledge	24-29 year	male	1	1	1	2	1	1	2	1	1
QA	24-29 year	female	3	2	1	1	3	3	2	3	2
QA	30-35 year	female	2	1	1	1	3	1	2	2	1
IT Expert	24-29 year	female	1	1	1	1	1	2	1	1	1
IT Expert	24-29 year	male	1	1	1	1	2	1	1	2	1
IT Expert	30-35 year	female	1	1	1	1	3	1	2	2	2
IT Expert	24-29 year	female	1	2	1	2	1	2	2	1	1
IT Expert	24-29 year	female	1	2	1	1	2	1	2	1	3
Total			24	26	24	26	33	25	30	32	29
Average			1.4	1.5	1.4	1.5	1.9	1.5	1.8	1.9	1.7

Figure 47: Observation Summary – Overall Impression