SOFTWARE REQUIREMENTS SPECIFICATION

for

<FinGer Shopping>

Version 1.0 approved

Prepared by <Rl>

<第11組>

June 15, 2018

Contents

1 Introduction 5
1.1 Purpose . . . 5
1.2 Document Conventions . . . 5
1.3 Intended Audience and Reading Suggestions . . . 5
1.4 Project Scope . . . 5
1.5 References . . . 5

2 產品描述 6
2.1 產品願景 . . . 6
2.2 產品功能 . . . 6
2.3 目標族群 . . . 6
2.4 開發環境 . . . 6
2.5 程式架構 . . . 6
2.6 User Documentation . . . 10
2.7 Assumptions and Dependencies . . . 10

3 External Interface Requirements 11
3.1 User Interfaces . . . 11
3.2 Hardware Interfaces . . . 11
3.3 Software Interfaces . . . 11
3.4 Communications Interfaces . . . 11

4 系統特色 12
4.1 輕鬆的購物方式 . . . 12
4.1.1 即時性 . . . 12
4.1.2 便利性 . . . 12
4.2 舒適的介面 . . . 12
4.2.1 暖色系的色彩 . . . 12
4.3 產品資訊透明化 . . . 12
4.3.1 評分系統 . . . 12
4.3.2 留言板的設計 . . . 12

5 Other Nonfunctional Requirements 13
5.1 Performance Requirements . . . 13
5.2 Safety Requirements . . . 13
5.3 Security Requirements . . . 13
5.4 Software Quality Attributes . . . 13
5.5 Business Rules . . . 14

6 Other Requirements 15
6.1 Appendix A: Glossary 15
6.2 Appendix B: Analysis Models 15
6.3 Appendix C: To Be Determined List 15

Revision History

Name	Date	Reason For Changes	Version
21	22	23	24
31	32	33	34

1 Introduction

1.1 Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

1.2 Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

1.3 Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

1.4 Project Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2 產品描述

2.1 產品願景

在這個網路蓬勃發展的時代，人民對於使用網路可以說是隨手可得。而藉由我們實做出來的交易平台，希望能提供使用者一個簡單且便利買賣 的環境，不僅如此，買家可以輕鬆方便的擔任商品的供給與需求方，更提透明的評分機制，讓使用者可以清楚了解商品的資訊，建立一個隨時隨地都能隨手做買賣的輕鬆便利交易制度。

2.2 產品功能

FinGer Sopping是一套讓使用者可以同時具備商品的供給與需求方的交易平台。擔任賣家可以透過新增商品同時上傳該商品的圖片、制訂商品價格來銷售自己的商品，並且將會收到來自買方的訂單需求通知；而擔任買家的一方，則可以透過流覽商品來進行選購，若有疑問可以透過商品的評價、或是利用留言版的功能來詢問賣家或其他使用者相關的商品資訊，購買後還能做後續的商品狀態的追蹤。

2.3 目標族群

FinGer Sopping成立的宗旨主要爲讓任何對銷售有興趣的使用者都能賣出自己的商品，平台提供賣方一個輕鬆便利提供商品的制度，達到人人都能在網路上當起大老闆；對於賣方則鎖定對網路相當熟悉的年輕人以及對於購買有強烈欲望的家庭主婦。

2.4 開發環境

此專案主要由 C# 中的 Window Form 來作開發環境，並且以 mongodb 做爲資料庫，來儲存用户的相關資料，成品說明則是由LaTex撰寫而成。

2.5 程式架構

使用MongoDB做爲資料庫，並統一用BaseDB作爲存取資料庫的基礎，再依照各頁面不同的需求來新增不同的class，來在相對應的collection進行一個存取的動作。

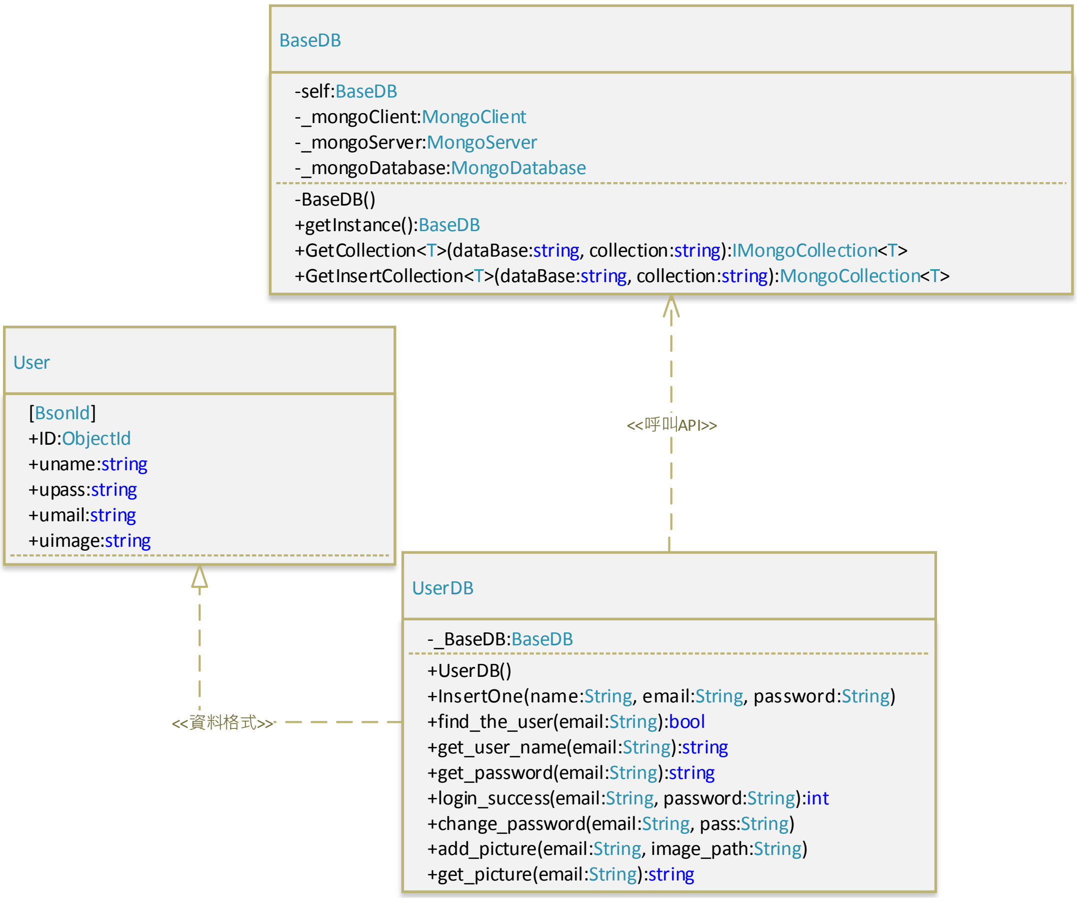
BaseDB
-self:BaseDB
-_mongoClient:MongoClient
-_mongoServer:MongoServer
-_mongoDatabase:MongoDatabase
-BaseDB()
+getInstance():BaseDB
+GetCollection<T>(dataBase:string, collection:string):IMongoCollection<T>
+GetInsertCollection<T>(dataBase:string, collection:string):MongoCollection<T>
User
[BsonId]
+ID:ObjectId
+uname:string
+upass:string
+umail:string
+uimage:string
<<呼叫API>>
<<資料格式>>
UserDB
-_BaseDB:BaseDB
+UserDB()
+InsertOne(name:String, email:String, password:String)
+find_the_user(email:String):bool
+get_user_name(email:String):string
+get_password(email:String):string
+login_success(email:String, password:String):int
+change_password(email:String, pass:String)
+add_picture(email:String, image_path:String)
+get_picture(email:String):string

Figure 2.1: UserDB用來處理新增、修改使用者登入到資料庫。

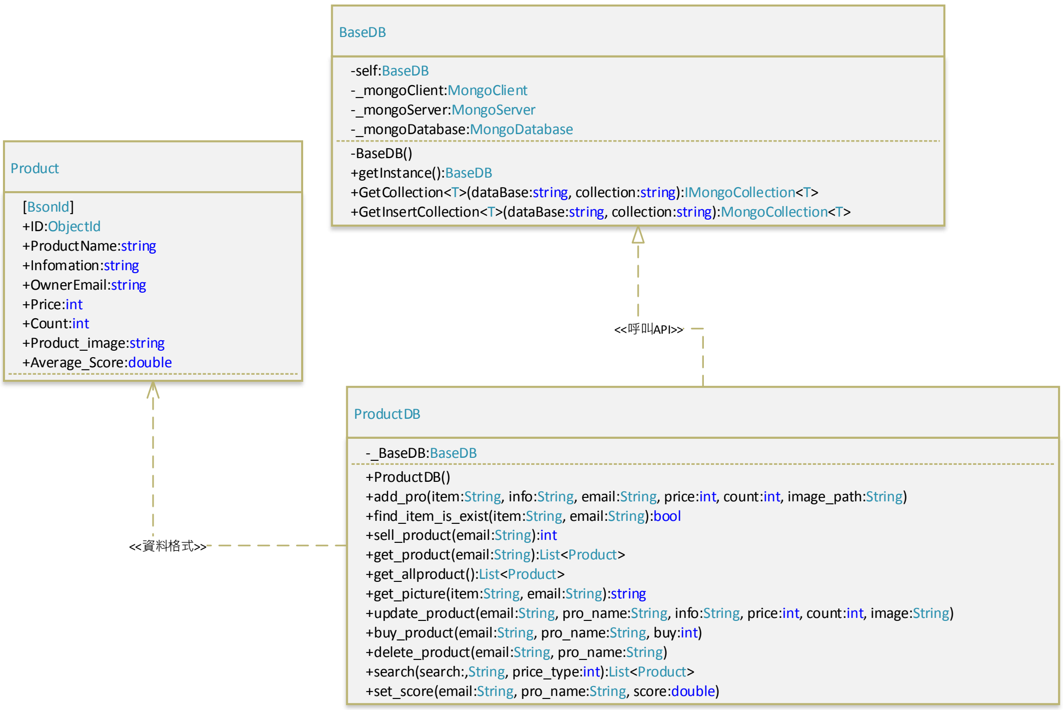
BaseDB
-self:BaseDB
-_mongoClient:MongoClient
-_mongoServer:MongoServer
-_mongoDatabase:MongoDatabase
-BaseDB()
+getInstance():BaseDB
+GetCollection<T>(dataBase:string, collection:string):IMongoCollection<T>
+GetInsertCollection<T>(dataBase:string, collection:string):MongoCollection<T>
Product
[BsonId]
+ID:ObjectId
+ProductName:string
+Infomation:string
+OwnerEmail:string
+Price:int
+Count:int
+Product_image:string
+Average_Score:double
<<呼叫API>>
<<資料格式>>
ProductDB
-_BaseDB:BaseDB
+ProductDB()
+add_pro(item:String, info:String, email:String, price:int, count:int, image_path:String)
+find_item_is_exist(item:String, email:String):bool
+sell_product(email:String):int
+get_product(email:String):List<Product>
+get_allproduct():List<Product>
+get_picture(item:String, email:String):string
+update_product(email:String, pro_name:String, info:String, price:int, count:int, image:String)
+buy_product(email:String, pro_name:String, buy:int)
+delete_product(email:String, pro_name:String)
+search(search:,String, price_type:int):List<Product>
+set_score(email:String, pro_name:String, score:double)

Figure 2.2: ProductDB用來對商品新增、修改、刪除。

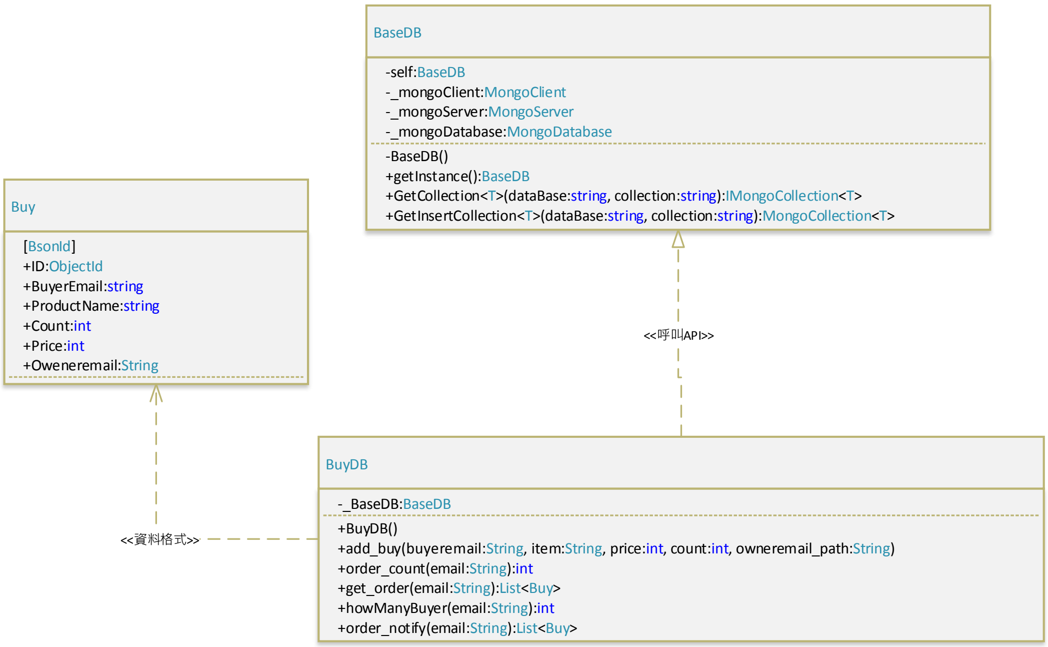
BaseDB
-self:BaseDB
-_mongoClient:MongoClient
-_mongoServer:MongoServer
-_mongoDatabase:MongoDatabase
-BaseDB()
+getInstance():BaseDB
+GetCollection<T>(dataBase:string, collection:string):IMongoCollection<T>
+GetInsertCollection<T>(dataBase:string, collection:string):MongoCollection<T>
Buy
[BsonId]
+ID:ObjectId
+BuyerEmail:string
+ProductName:string
+Count:int
+Price:int
+Oweneremail:String
<<呼叫API>>
<<資料格式>>
BuyDB
-_BaseDB:BaseDB
+BuyDB()
+add_buy(buyeremail:String, item:String, price:int, count:int, owneremail_path:String)
+order_count(email:String):int
+get_order(email:String):List<Buy>
+howManyBuyer(email:String):int
+order_notify(email:String):List<Buy>

Figure 2.3: BuyDB用來對儲存商品購買紀錄。

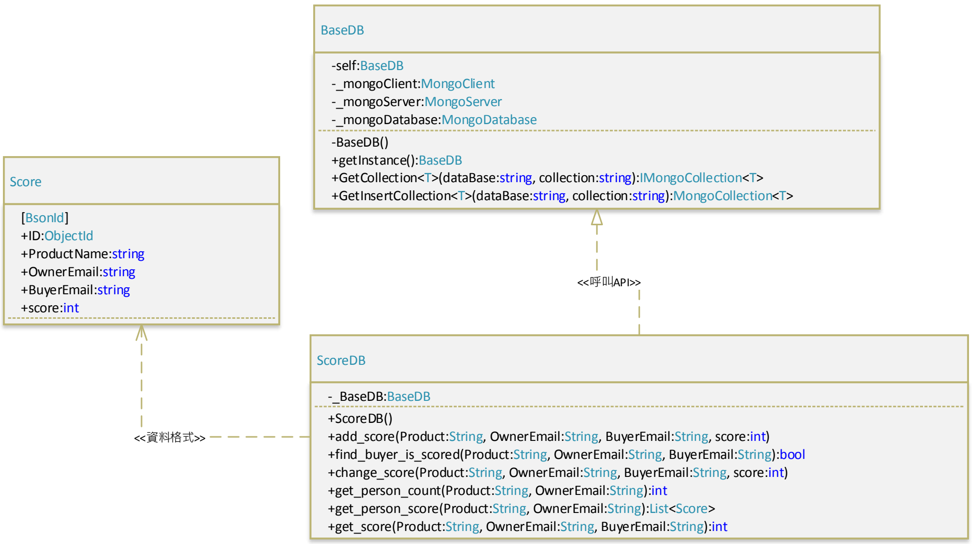
BaseDB
-self:BaseDB
-_mongoClient:MongoClient
-_mongoServer:MongoServer
-_mongoDatabase:MongoDatabase
-BaseDB()
+getInstance():BaseDB
+GetCollection<T>(dataBase:string, collection:string):IMongoCollection<T>
+GetInsertCollection<T>(dataBase:string, collection:string):MongoCollection<T>
Score
[BsonId]
+ID:ObjectId
+ProductName:string
+OwnerEmail:string
+BuyerEmail:string
+score:int
<<呼叫API>>
<<資料格式>>
ScoreDB
-_BaseDB:BaseDB
+ScoreDB()
+add_score(Product:String, OwnerEmail:String, BuyerEmail:String, score:int)
+find_buyer_is_scored(Product:String, OwnerEmail:String, BuyerEmail:String):bool
+change_score(Product:String, OwnerEmail:String, BuyerEmail:String, score:int)
+get_person_count(Product:String, OwnerEmail:String):int
+get_person_score(Product:String, OwnerEmail:String):List<Score>
+get_score(Product:String, OwnerEmail:String, BuyerEmail:String):int

Figure 2.4: ScoreDB用來紀錄不同使用者對商品的評分。

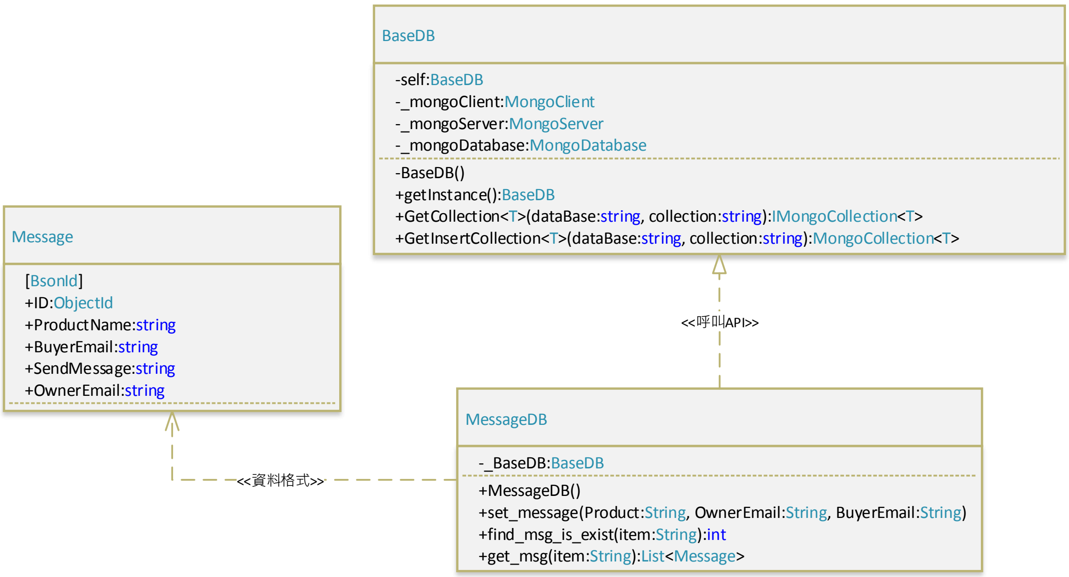
BaseDB
-self:BaseDB
-_mongoClient:MongoClient
-_mongoServer:MongoServer
-_mongoDatabase:MongoDatabase
-BaseDB()
+getInstance():BaseDB
+GetCollection<T>(dataBase:string, collection:string):IMongoCollection<T>
+GetInsertCollection<T>(dataBase:string, collection:string):MongoCollection<T>
Message
[BsonId]
+ID:ObjectId
+ProductName:string
+BuyerEmail:string
+SendMessage:string
+OwnerEmail:string
<<呼叫API>>
<<資料格式>>
MessageDB
-_BaseDB:BaseDB
+MessageDB()
+set_message(Product:String, OwnerEmail:String, BuyerEmail:String)
+find_msg_is_exist(item:String):int
+get_msg(item:String):List<Message>

Figure 2.5: MessageDB用來紀錄不同商品使用者對其的留言。

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3 External Interface Requirements

3.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4 系統特色

4.1 輕鬆的購物方式

4.1.1 即時性

賣家可以在 FinGer Sopping 隨時新增商品，在有使用者購買其商品時，能提供即時的買賣資訊和通知；買家也可以隨時選取自己想要的商品進行購買，並且即時的追蹤商品的狀態。

4.1.2 便利性

在任何有網路的地方，都能隨時當起買家和賣家，只要輕鬆按下一鍵，就能在賣家和買家之間能夠自由切換自己的角色。

4.2 舒適的介面

4.2.1 暖色系的色彩

根據研究，由於受到色彩的視覺刺激，而在思維上有所影響。而我們在 FinGer Sopping 上大量採用暖色系的顏料，人類對於暖色系的色彩會感到興奮感，進而提升買家購買的慾望，而對於暖色系的色彩也比較不會對人類的視覺感到疲憊。

4.3 產品資訊透明化

4.3.1 評分系統

藉由買家購買的感受度，來給商品評分，讓好的賣家能夠得到更多的回饋，也能讓使用者能夠挑選較優質的商品。

4.3.2 留言板的設計

若買家對於商品存在任何疑問，可以透過 FinGer Sopping 提供使用者的留言版來詢問賣家產品的相關問題，也能讓其他想購買此商品的買家能夠在留言板上尋找相關的答案或給予相關的意見。

5 Other Nonfunctional Requirements

5.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

5.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

5.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

5.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

6 Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

6.1 Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

6.2 Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

6.3 Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>