DUBLIN INSTITUTE of TECHNOLOGY
Institiúid Teicneolaíochta Bhaile Átha Cliath

Contingency Management System
Interim Report

DT228
BSc in Computer Science

Leslie Ducray
C10327999
Supervisor: Richard Lawlor

School of Computing
Scoil na Ríomhaireachta

Abstract

The purpose of this report is to document the processes underpinning the research, design, development and implementation of a Contingency Management web application. Though the application is developed to modify drug misuse, the fundamental activities behind this application could be used to promote many other forms of behaviour change, via the underlying principles of operant reinforcement and behavioural economics. Contingency Management is a highly effective, evidence based, incentive centred treatment for problem behaviours. This approach is most commonly used to strategically treat drug and alcohol addiction through offering tangible and meaningful rewards that are contingent upon agreed and targeted behaviour change. Whilst the Contingency Management approach is clinically highly effective the adoption of an automated web application will enable a determination of complex applied reinforcement criteria in 'real time' and improve communication between treatment staff and service users. The proposed system aims to automate a current paper-based system, removing the tedious and laborious act of manually processing a service user's treatment progression. As per international best practice advice this system aims to target two objective behavioural aspects: engagement in clinical activities and drug use.

A major benefit main of this system is the ability to minimise human error and bias in creating an automated approach to recording and calculating the patient's 'real time' reward status and entitlements (normally vouchers to a calculated monetary value and/ or "take away" medication privileges) in a far less laborious fashion than which is currently employed in many treatment facilities. A second benefit is the potential opportunity to generate reports about patient progress and accumulate outcomes research data, an important consideration in 'evidence based' clinical practice. This application could be employed in a range of facilities including but not limited to outpatient drug treatment centres, residential programme, prisons, the military and juvenile detention centres etc.

Table of Contents

Table of Tables

Table of Figures

Chapter 1 Introduction and Background

According to The European Monitoring Centre for Drugs and Drug Addiction (EMCDDA) in 2012 there were at least 1.2 million people in Europe receiving treatment for illicit drug use, of which the majority (630 000) receive treatment for opioid abuse (1).

With regards to the supply and use of illicit drugs in Europe Cecilia Malmström, the European Commissioner for Home Affairs declared,
"*We are faced with an ever more complex stimulant market and a relentless supply of new drugs which are increasingly diverse. The fact that over 70 new drugs have been detected in the last year is proof in itself that drug policies need to stay on target*" (1).

In Ireland, the Health Research Board (HRB) reported that in 2010 4,930 patients entered into treatment for opioid dependence, mainly heroin and that 1,893 received treatment for cannabis use.
(1).

The Drug Treatment Centre Board (DTCB) , Ireland's oldest opiate addiction treatment centre based in Dublin, reported in their 2011 annual report that they treated 864 individuals for drug addiction, of which 68% were male (2). This report also noted the homeless population receiving treatment for addiction had risen to 233 in 2011 from 181 in 2007. This agency also advised that the (Irish) National Drug Analysis Laboratory conducted 1,187,926 drug screening tests in 2011 via urine, blood and oral fluids.

Further statistics furnished by Merchants Quay Ireland, a non-profit drug service provider in Ireland, indicated this body received 20,847 visits from 3,634 service user to make use of their needle exchange service in 2012 (3). This was an increase of over 2,000 visits from 2011. Tony Geoghegan, the Chief Executive for Merchants Quay Ireland, declared in their annual report "*Demand for our homeless and drugs services is growing, yet finances are contracting*" (3).

The National Institute for Health and Care Excellence (NICE), a highly influential body within the Department of Health in the United Kingdom published a document in 2007 outlining their recommended approaches for drug treatment. In particular the Contingency Management approach was highly endorsed as a psycho-social intervention for drug misuse (4).

According to the prestigious NICE Guidelines and other authorities Contingency Management is considered a psychosocial treatment system with a very high degree of scientific evidence base (5) (6) (7) (4).This approach aims to treat patients by offering incentives to reward or reinforce various types of behaviour change. Contingency Management, In particular this approach has been shown to be very beneficial to a population living extreme circumstance (e.g. the homeless, pregnant woman, people with mental illness) (6).
Chapter two will highlight more research into Contingency Management, the background of the approach and previous automated adaptions.

The motivation for this project is to create an automated generic web application for staff working at a drug treatment facility which will allow them to admit, manage and monitor the progress of service users engaged within a Contingency Management program and automatically apply applicable reward and reset rules based on a defined criteria. The primary function of the system will be to apply a reinforcement or reward (Contingency) process to service users based on attainment of negotiated or agreed targeted behavior change (i.e drug use, attendance, participation in therapeutic activities offered etc.). The criteria applies logic based on objectively verifiable results from a clients' laboratory test data or behaviors such as clinic or meeting attendance etc. This particular implementation of a contingency management system will group service users of the treatment-based center into two different progress streams, based on predetermined criteria (stability, behavior, extent of drug use). The two streams are identified for convenience as a 'High Support' stream and a 'Progression' stream. The reward protocol of each stream is governed by different criteria. Service users can also be penalized for breaches of their agreed treatment contract including drug use, failure to attend agreed meetings without good cause and other relapsing behavior via a loss of opportunity to receive certain credits and /or a 'reset' to a less 'valuable' reward schedule or protocol. As the system is based on the systematic and structured application of earned incentives, service users will accumulate these rewards. A large part of the application will be to record the transactions of service users. The transactions functionality will feature a balance of the clients earned rewards and eligibilities, 'sanctions' (loss of credits and 'resets') and a history of credits earned and withdrawn. This particular approach will only monitor two predetermined types of behaviours as per international best practice.

The system will need to attain and store information on each service user including

- Appropriate identifying and demographic data
- Meetings attended / general attendance
- Engagement in target therapeutic activities
- Drug screening tests for various categories of drug use (e.g. Opiate, benzodiazepine, stimulant, cannabis, alcohol etc.)
- "Take Away medication" and other treatment privileges/ eligibilities
- Service User's Treatment Stream
- Reward history
- Rewards/Credit account – balance, history and transactions.
- Notes from staff

The web application will be developed in Java, JSP, JavaScript and SQL primarily and will rely upon a rule based expert system (Drools) to govern the underlying Contingency Management criteria.

The primary objective of this application is to provide a web application user interface for staff of a treatment center employing the Contingency Management approach. The web application will be required to correctly and quickly manage data requests and responses between client (browser) and server. The system functionality will need to correctly calculate the outcomes of the rule based system based on user input from attendance meetings and tested samples. Chapter 4 will cover the architecture and implementation process of the full system scope.

Chapter 2 Research

Contingency Management

Research into the proposed project begun by studying and understanding the principles of Contingency Management, its application and its target environment. Contingency Management is a psycho-social treatment approach which focusses on the consequences of a person's actions by rewarding or depriving them of incentives, based on their behaviour (8) (7) (5) (6). The Contingency Management process has a set of defined rules which govern the reward process. Contingency Management is a form of drug and alcohol misuse treatment, but its logic abstracted from content could be applied to many other behavioural economic purposes (e.g. customer loyalty schemes, public health initiatives, macro- level driver or road safety schemes, tax compliance). The Contingency management approach reflects the behavioural economics theory as by offering incentives, the abuse of substances will decline, becoming less attractive as abstaining becomes more attractive (9).

Contingency Management for substance and alcohol abuse is mostly administered in treatment centres (drug treatment centres, prisons, juvenile detention centres etc.).

The rules and logic to follow are adapted from existing paper based implementations of the Contingency Management approach. Contingency Management is applied to each new transaction of

- The service user's attendance (attended, missed or missed with valid reason)
- The result of their substance tests (negative or positive for the presence of the tested substances).

These outcomes determine

1. The amount of points or credits a service user earns which is added to their balance. This balance can be converted to shopping centre vouchers which have a monetary value, personal to holder and which can be redeemed for goods excluding tobacco and alcohol.
2. Their eligibilities earned/lost.
3. Their stream progression/regression.
4. Their date to be clean by ('Date to Clean').

For a service user to withdraw their earned credits they need to have a certain pre-agreed minimum amount accumulated (in this instance €20). They also need to have tested negative (i.e. no opiates or cocaine) in their most recent substance tests for a 'Financial User' to approve the withdrawal of their earned credits.

Through communication with a clinical psychologist working at a Dublin drug treatment centre which employs a paper based implementation of the Contingency Management approach, the following flow of process was elicited and discussed. This system is based on the scientific literature outlining those features or principles considered to be core to an effective Contingency Management scheme which include: immediacy; an increasing or escalating schedule of reinforcers; progression or regression between ‘streams’ or ‘levels’; a “reset function” and receipt of contingencies (or rewards) which are personally meaningful or rewarding to the participant.

When a service user (client/patient) is first admitted into the Contingency Management program, they are assigned to an agreed upon treatment stream. Generally the initial treatment stream is the foundation or entry level stream, in this instance termed the ‘High Support’ stream. Ideally when the patient progresses in their treatment and stability is established they progress upwards to the next ‘level’ or ‘stream’. Each stream offers a different magnitude of rewards and likewise a different magnitude of penalty for misbehaviour. The service user’s details are elicited and recorded.

Service users of the system are ideally critiqued on no more than two different target behaviors. The two explicitly target behaviors that will be addressed in the development of this Contingency Management system is 1) Engagement (participation in arranged meetings and therapeutic activities) and 2) Drug Use (as measured by the service user’s regular drug screening tests).

The results of these drug tests impact upon the progression and regression of service users through the two streams. If a service user consecutively tests drug negative (no drugs) they will progress from the ‘High Support’ stream to ‘Progression’ stream. Also with each drug negative test provided per schedule, the service user incrementally accrues more reward points, which results in them receiving a reward of a higher total magnitude. This is termed as an escalating schedule of reinforcers. In situations where service users again test positive for drugs after periods of abstinence their incrementally increased ‘reward point accumulator’ is reset to its reset value (0). Users do not lose points earned, but they will need to re-build their accumulators to be rewarded more. Likewise if a service user consecutively tests positive or lapses after a period of abstinence they will regress back to ‘High Support’ stream and/or be issued deadlines to test negative (”Date to be clear”). These dates are issued to give one weeks’ notice of the testing expectation. If the service user fails to meet their ‘Date to be clear” test, they are issued another date in a weeks’ time. These “Date to be clear” targets are progressively issued as a card system (for example progressing through ‘blue’, ‘yellow’ and ‘red cards’. Once the ‘Date to be clear’ associated with the ‘red card’ has failed to be met the patient will be invited to re-assess their desire and motivation to continue within the Contingency Management program.

Attending specific treatment meetings (e.g. Key Worker, Counsellor, Social Worker meetings etc) also allows the service user to earn a capped number of reward points.

Through personal correspondence with Dr. Nancy Petry, leading international authority in contingency management and the developer of the low-cost contingency management treatment approach (7); the project proposal was discussed.

With reference to previous attempts to develop automated Contingency Management systems, Dr. Petry had this to say “*Several groups have attempted to develop automated CM programs, but none have been widely implemented in practice*.” Dr. Petry went on to discuss the previous short falls of previous implementations she has been exposed to and explained corrective measures at trying to better ensure a correct implementation.

"*Some of the problems have been that each clinic seems to want to implement CM in its own way, and the programs developed thus far have not been able to accommodate issues such as different reinforcement magnitudes and types (vouchers vs. prizes vs. stars exchangeable for items), different attendance structures (which is less of an issue for opioid replacement treatments relative to outpatient care), different durations of CM interventions, and excused absences.*" Many of these issues had been taken into consideration in the initial proposed approach, but the comments stated offered more definition and confirmation for the planned implementation. Dr. Petry continued on to state issues regarding valid reasons for being absent from arranged meetings and sample taking, "*Because the evidence-based CM programs have escalating reinforcers for sustained abstinence, the programs developed to date have not been able to incorporate this feature appropriately given the real world issues that arise (e.g., in terms of missed samples for legitimate reasons etc.).*". Previous implementations have enforced the Contingency Management criteria too rigidly without offering an override or exception to legitimate reasons of absence, thus penalizing the service user 'unfairly'.

Application Medium and Platform Research

The decision to implement a web application opposed to a native application or an application for a mobile device was decided based on the following considerations:

- **Where will the users use the application?**
  Typically users of the system will be based in clinic offices, laboratory environments, research centres or even mobile treatment centres (modified treatment busses) where dedicated staff members can record test results/notes/attendance. The interactions between service user and treatment staff will require a level of privacy.

- **Who will the typical users be?**
  Typically the users of the system will be members of a clinical team such as doctors, psychologists, counsellors, or Key Workers who interact directly with the service users or process their test results.

- **What medium/technology will the users have?**
  In a treatment centre or correctional facility, typically the administration staff would have a personal computer/laptop for keeping records. In many cases, due to the one-on-one nature of the interaction between a service user and a member of treatment centre staff, information would be taken down during the meeting and this could be via a paper notes, tablet or other mobile devices.

- **What will be the most accessible means of distributing the system?**
  Due to the nature of the confidential and sensitive information stored, the system will need to be totally secure and meet all data protection and other legislative requirements. A web application hosted on a local and secure

network would appropriately cater for this requirement. Web applications can be run and rendered on almost every platform that has a web browser installed, this includes smartphones and tablets. The user interface can be easily and efficiently designed using HTML5 and can be enriched with native client side technologies. User sessions and login times can be configured to maintain secure access. Disadvantages of developing the application for a specific platform such as IOS or Android mobile devices would render a large portion of the target platforms incompatible, as these technologies are specific to their applicable devices. IOS applications require an Apple Mac device to be developed on with XCode and IOS SDK installed and an Apple device to be run (10), this is not a feasible option for the development of this application, as this system is intended for health services where a financial budget is restricted.

Publishing any developed applications to the Apple Store also requires developer's fee and restrictive licencing. While Google Android mobile devices demand no developer's fees for publishing to their App Store, making them financially more appealing, they are still restrictive to the compatibility of the devices which can run the applications. Web applications designed to be received and rendered through the users browser is the most feasible option for this particular application as where the user primarily works from their own office with their desktop/ laptop or need to enter in the information from their mobile device, the web application can be accessible regardless of their device and/or platform. As with web applications the application is deployed to a host; there is no need for it to be published to application stores, which will enhance re-deployment time for updates, changes or bug fixes. Should there be changes made, the users will not need to download or install new versions. As the system will be hosted on a local area network, there are no hosting costs.

HTML5, JavaScript and CSS technologies can be employed by the application to enhance the user's view of the application on mobile devices using adaptive or responsive web design techniques of displaying and sizing the web page content.

Disadvantages of developing a web application could include compromising the browser's performance due to requests and response wait times. The nature of this application does not require an intense low latency interactivity so the user should not be noticeably affected or disadvantaged by delays. Another issue that can arise with a web application is the desire or need to continue working off-line or due to a lack of connection. As mentioned previously, the access and use of the web application will be limited to the confines of the local network which ensure a more secure and deliberate limited external access. As the objective hosting of this application is intended to be hosted on a local area network, an external internet connection (to the World Wide Web) is not a necessary requirement.

- **How will this application help?** If implemented correctly, this application will provide enhanced efficiency, accuracy and labour saving to clinical staff of a treatment centres, aiding them by bypassing the act of having to manually apply the Contingency Management criteria and also allowing the provision of feedback as to “rewards or credit balance accrued” in near real time. It will provide treatment providers with a much more efficient record keeping facilities and a wealth of data for scientific and peer reviewed research. In an economic climate where Health care funding comes with budget constraints, a system that will enhance productivity of staff could prove very beneficial.

Database

As the system will rely quite heavily on data storage, a reliable and efficient database management system is needed. The design and implementation of the database is required to be quite sophisticated in its associations. As the business logic of the web application is firmly based on the governing rules of Contingency Management, implementing the database to dynamically aid the rule based system will greatly improve the maintainability, consistency and dependency of the system.

The structure and nature of the data storage required for the proposed Contingency Management system is relatively rigid and tabular in format. The associations between table entities will serve the database systems primary function for storing and retrieving data. Any change or volatility of the data model/schema is unlikely expected, as the associations between table entities will be closely based on the principles of Contingency Management and information required, and the entity attributes will be designed to allow a certain degree of dynamics.

With this statement in mind, the need for the likes of a NoSQL database such as Mongo DB would serve little benefit. NoSQL databases employ a less constrained data model that allows for retrieval and storage of data in a more flexible manner. Where relational models separate data into various associated tables joined by identifiable keys, document orientated NoSQL databases such as Mongo DB aggregate the data into one particular record for storage, thus removing the need for entity relationships (11).

A relational database such as MySQL would suit the nature of a Contingency Management system, offering a simple and set approach for storing, querying and retrieving data. MySQL has the power to employ the option of data warehousing which could be an expansion of analytical and report generations for the application in the future. MySQL is one of the leading popular open source relational database managers, offering in addition a MySQL server and the workbench development kit for design and implementation. As the scope of the application by design will be limited to the local network of treatment facility, where the amount of traffic to and from the database will remain relatively low and the expected wish to scale the database is non-existing, a MySQL database is ideally suited.

Web Technology

Sun’s J2EE, Microsoft ASP.NET and Linux/Apache/MySQL/PHP (LAMP) are the three leading and most popular Web development platforms. When comparing these three platforms to decide on the most suitable and applicable, it is very plain to see the similarities and competing functionalities at a high level.

Table 1: Comparison of web development platforms (12)

Criteria	J2EE	ASP.NET	LAMP
Database Compatibility	Multiple	Multiple	Multiple
Accepted Platforms	Multiple	Windows only (12)	Multiple
Server Compatibility	Multiple	Windows Server only (12)	Multiple
Frameworks	Many to choose from	Many to choose from	Many to choose from
Licensing cost	Free	Free	Free
Support, Documentation and online resources	Yes	Yes	Yes

Deciding on a web technology platform required that the platform could integrate correctly and reliably with the chosen database server which is a MySQL server. Java Database Connector (JDBC) forms a framework for communicating between the Java application and the MySQL server. These technologies are all open source and widely used and well documented.

Java Enterprise Edition (J2EE), an expansion of the standard Java platform allows for the inclusions of many pre-existing libraries. These are a large attraction as they have under gone thorough and various testing through various implementations. Such libraries will offer more reliability and productivity than having to re-develop the existing functionality (13). J2EE by default includes many APIs which are necessary for developing a web application such as the JDBC. J2EE also comes pre-set with the various associated file types such as Java Beans, Java Servlets and Java Servlet pages through Eclipse to be included in the developed application. J2EE was developed to aid developers in the design of large multi-tiered applications for reliable and secure distribution over a network (13).

Spring is a framework designed to architecturally structure the entire scope of the web application (14). The Spring Framework will aid in organizing the structure, abstraction and dependencies of the application's many components and java objects.

Rule Engine

Deciding on a rule based expert system to incorporate into the system was dependant on a few factors

- **Will it integrate well with the existing chosen technologies?**
  As the current web application technologies and languages decided are Java and MySQL, the middle wear needs to be able to interact with the select rules engine.
- **Is the software open source or is purchased licence required?**
  Initially JESS (Java Expert System Shell) was going to be the preferred option to incorporate into the system until Licensing was looked into. A personal en-

quiry was made to the Software Licensing Administrator at Sandia National Laboratories requesting information on obtaining a licence for this particular project.
An academic licence was attained under strict terms and conditions that the academic licence is restricted to only using JESS on a project specific basis for academic and non-proprietary and/or commercial research or development.

  Open source and Java compatible rule based engines were also researched and compared. 'Drools' is an open source forward chaining inference based rule management system that was decided on.

- **Is a rule engine necessary?** As the primary business logic of the system is focussed on applying criteria to know facts, a rule engine could be very effective in structuring and optimizing the calculation process. Using the rule base will be an optimal means of organizing the Contingency Management criteria.

Drools employ an enhanced version of the rete algorithm and can easily be incorporated with in a Java application (15). In addition to the rule engine, the Drools software also includes visual aids which illustrate the flow of the rules which will aid in the development. The Drools software can be a ‘plug-in’ for the Eclipse SDK.

Chapter 3 Design Methodologies

The design and implementation of this project began at a database level. The system is very dependent on the structure of the database entities and their associations. The database is firmly based around a Contingency Management paper-based system. There were multiple implementations of the database before further web application development could begin.

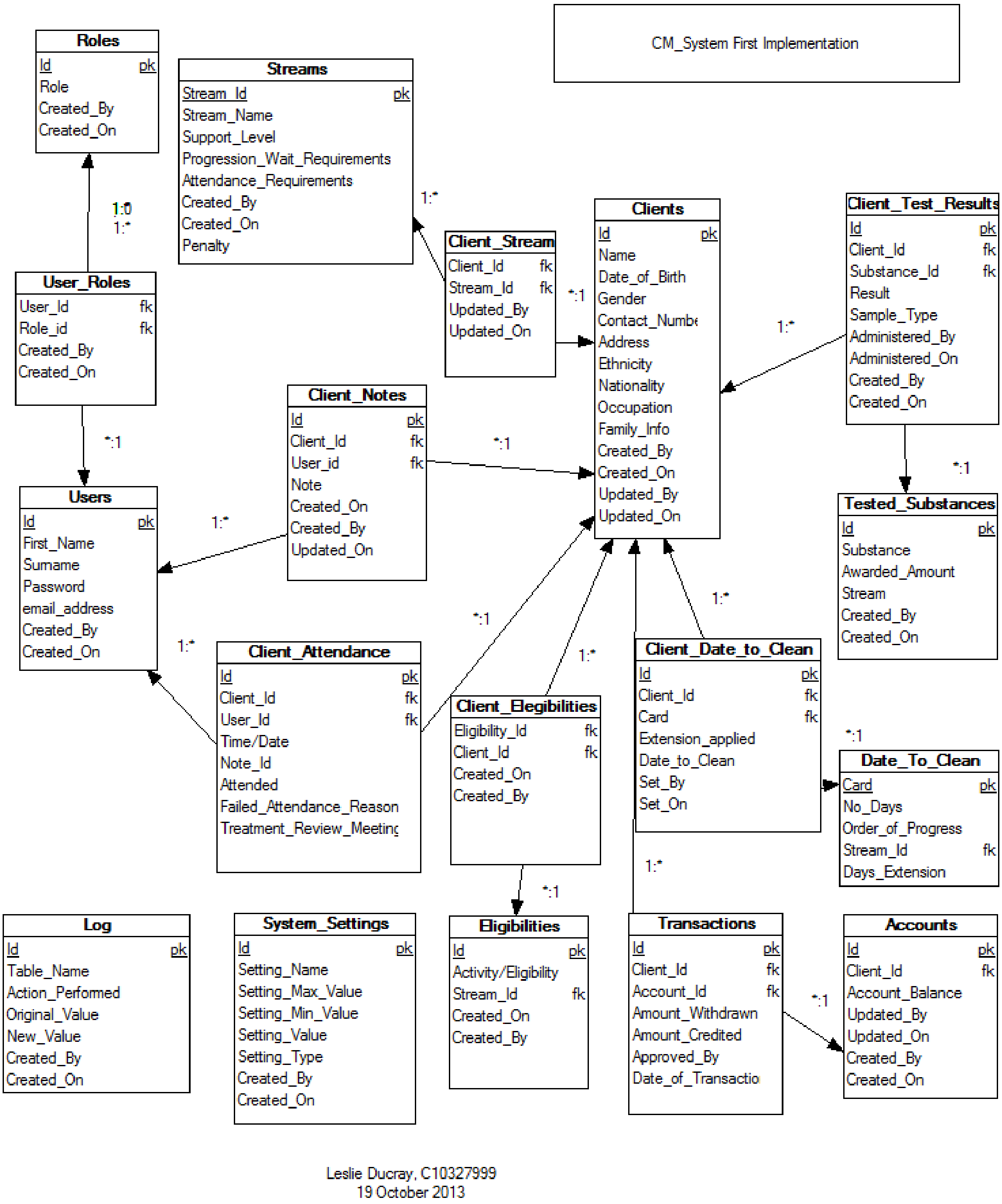

Figure 1: First database schema implementation

The first implementation of the database was a rough design model recording all the necessary attributes, keys required, and the relationships between entities. The first implementation included a full schema logging table. This was removed in later implementations. Instead each table record will store the user key and the date associated with its creation and this can alternatively be used as a log. All associations to the 'User' table are not recorded in the above diagram as the many associations with the 'User' table impairs the readability of the diagram. Figure 2, illustrates the associations between each entity more explicitly. Figure 3 expands on Figure 2, illustrating each entity's attributes. The design of the database schema is normalized to ensure the schema is more maintainable and the degree of dependency between entities is minimised.

The database schema is composed of sixteen tables catering for storage of all relative information. A summary of each table is as follows:

- User – The User table stores all static general information related to a member of staff of the treatment facility.
- Role – The Roles are permission statuses for the system.
- User Role – the joining association assigning a role to a user.
- Service User/ clients – This table represented as Service user or Clients stores all general and static information regarding a service user.
- Service User notes – this association table joins a service user and user where the user has recorded a note on the service user.
- Service User attendance – records a service user's meeting with a user (i.e. clinician). This table records if a service user fails to make an arranged meeting and whether they have a legitimate excuse (which will spare them penalization when Contingency Management rules are applied).
- Substance – recorded information regarding substances and their effect on the Contingency Management criteria.
- Service User Substance Result – This table forms an associative history for each service user and the results of their regular drug use screening tests storing their urinalysis result, date administered, associated points earned etc.
- Stream – this table stores general information regarding the Contingency Management streams- different levels of progression within the Contingency Management process.
- Service User Stream – The stream associated with a service user. Their current level of progression.
- Account – A service user's account stores the balance of their earning.
- Transactions – the records of all rewards earned/credits and withdrawals/debits made to a service user's account.
- Eligibility – the information regarding all Eligibilities – such as 'eligible to with draw' , 'outings/visits' , 'take outs'.

- Service User eligibility – an association of which service users have which eligibilities.
- Date to clean – this table stores the information regarding date to clean rules. ‘Date to Clean’ is a date set by a user of the system assigned to a service user to test negative in a substance test.
- Service user date to clean – A recorded history of service users and their assigned dates to be clean, if extensions were given, consecutive failures.

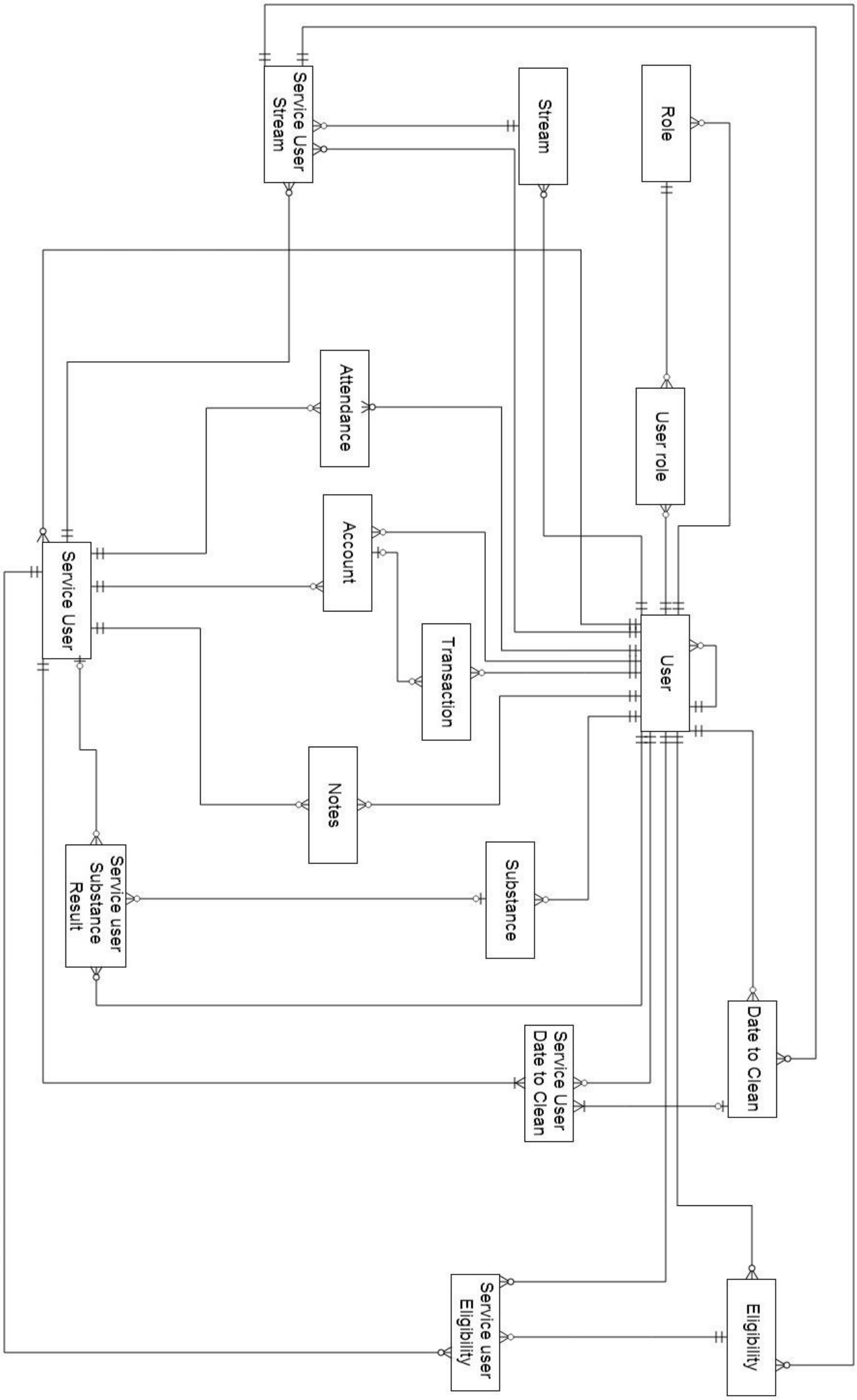

Figure 2: Full representation of associations between entities

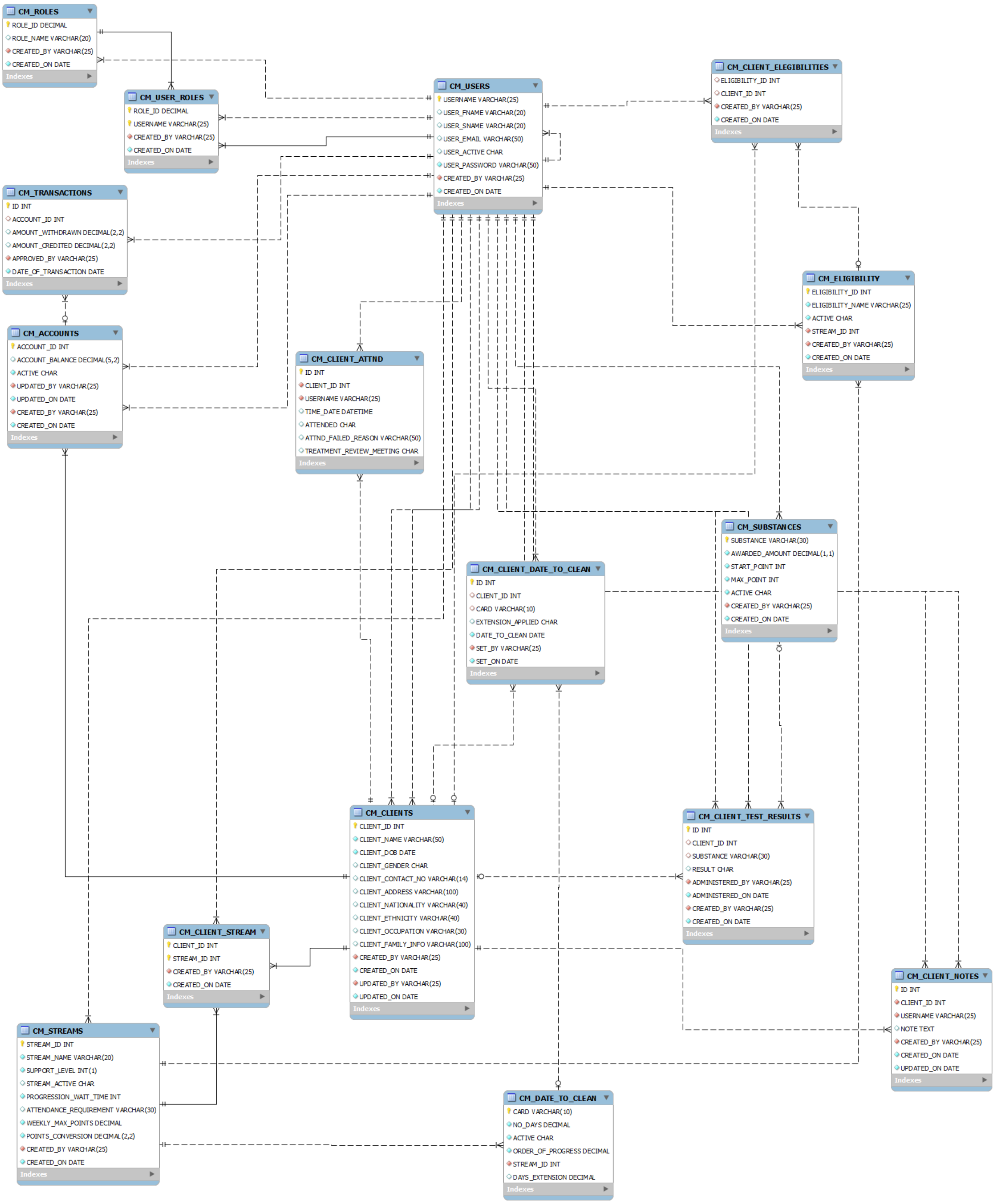

Figure 3: The current database schema.

Figure 3 has been reverse engineered using the SQL script files and MySQL Workbench Modelling tool.

Once the underlying database structure was of confident design, the design and planning began on the expectations of the web application. These expectations are represented in the use case featured in Figure 4. Figure 4 defines the three user (actor) roles of the system:

1. User, the default role of any active user of the system. This role permits a user to add new clients/service users, edit existing ones, record test results and attendance which in turn applies the contingency management criteria.
2. Financial User, a role assigned to a user permitting them to view the account of a service user and approve the withdrawal and/or assignment of rewards to a service user.
3. Admin is the administration role. This role permits a user to add/edit users of the system.

Users of the 'Financial' and 'Admin' role are permitted to all actions of the default user role.

The system actor forms the business logic and control of the application. The system will be required to:

- Establish a connection with the database.
- Respond to user requests.
- Validate user login and assign roles.
- Manage the user session.
- Process queries/updates/transaction statements to the database.
- Calculate and apply the Contingency Management rule criteria.

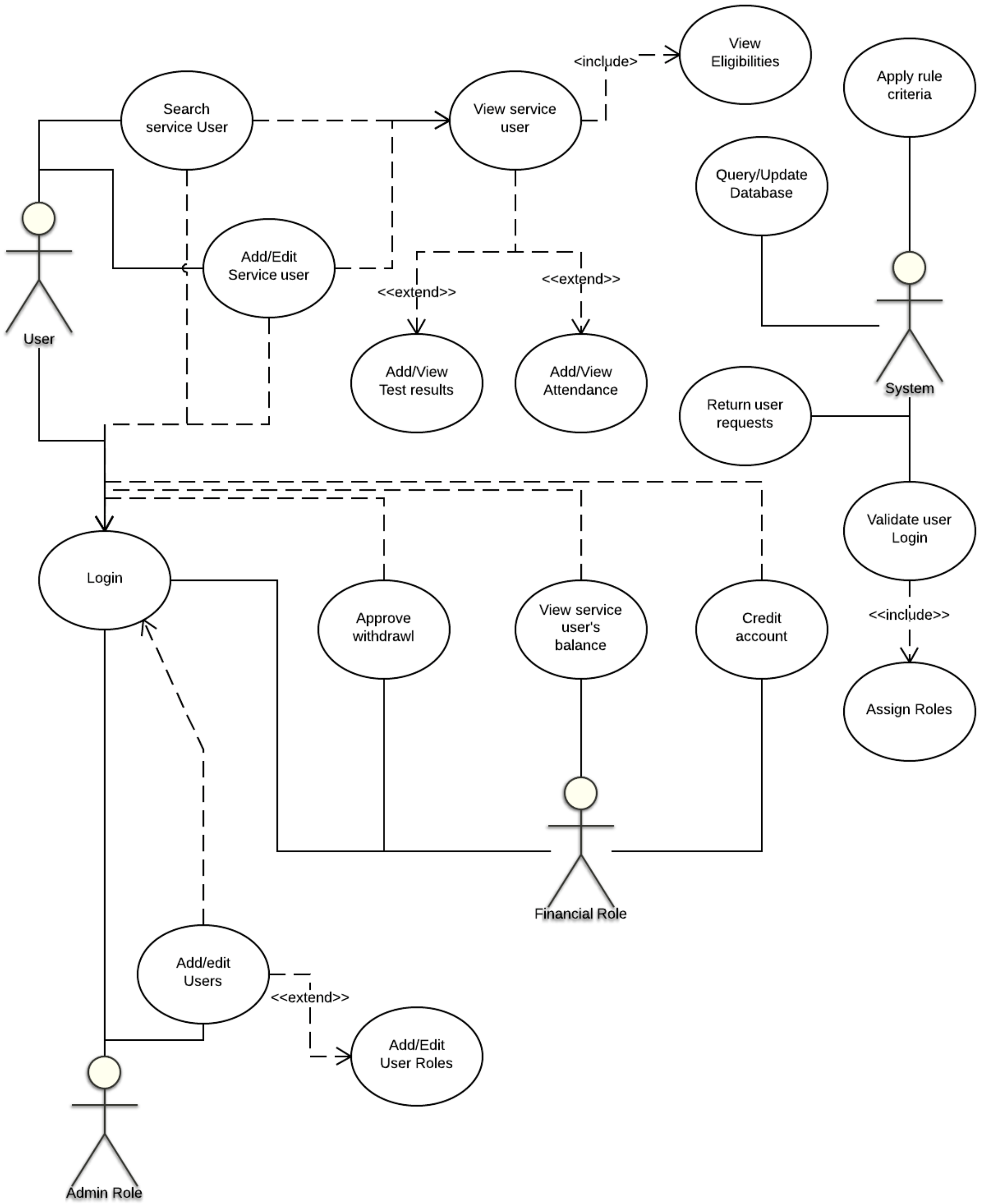

Figure 4: High level Use Case of the system

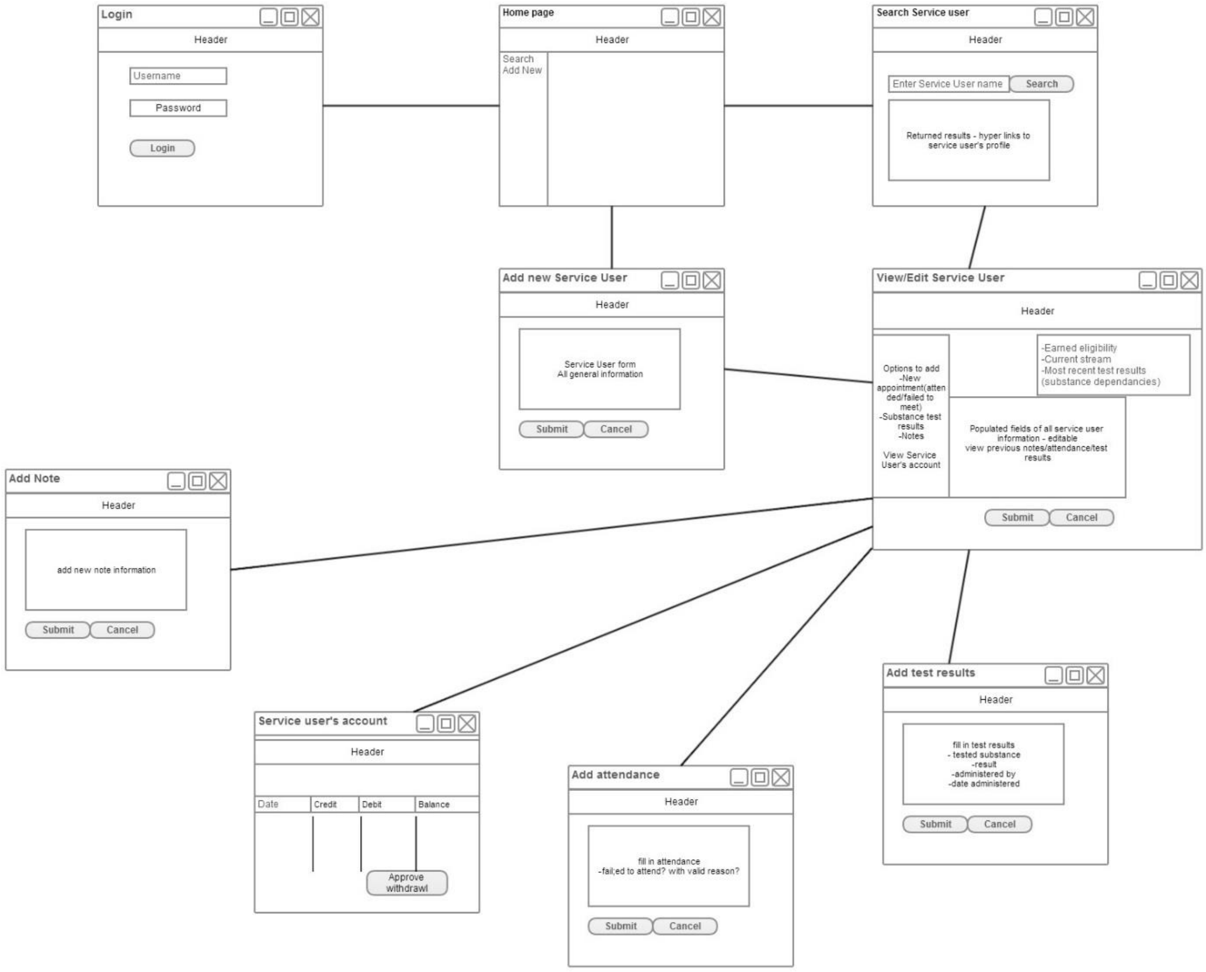

Figure 5: Front End Horizontal prototype

Figure 5 illustrates the conceptual layout of the web application and the flow between pages as presented to the user.

The design methodology chosen to guide the implementation of this application is aspects of the Agile design methodology and Scrum applicable to a single person development team. This methodology feels most suitable to the application, allowing the user to plan sprints/iterations (an independent aspect/module of the system to be developed), design and analyse, implement, test and evaluate *(16)*. The full system plan is divided up into independent sprints or iterations.

This allows an assessment to be performed at each iteration of a particular aspect of the system to review progress and priorities and to test and evaluate current implementations. The Agile and Scrum methodology encourages an incrementing implementation design and development process *(16)*. Using this methodology, the progress and implementation of the application can consistently be assessed, and each aspect of the system will be concluded instead of developing multiple incomplete aspects. Having a complete system sprint fully functioning will aid in testing and evaluation, allowing target users to review and provide feedback on a module for consideration, improvement and rectification.

With the general required functionality of the system established, the web application was divided into multiple independent sprints for design and development. These sprints will be discussed further in Chapter 4 Architecture & Development.

Chapter 4 Architecture & Development

The application is developed using Java technologies. J2EE is an enterprise expansion of the standard Java platform. Java technologies incorporate both the running platform/environment (which consists of a Java virtual machine) and the application programming interface (13).

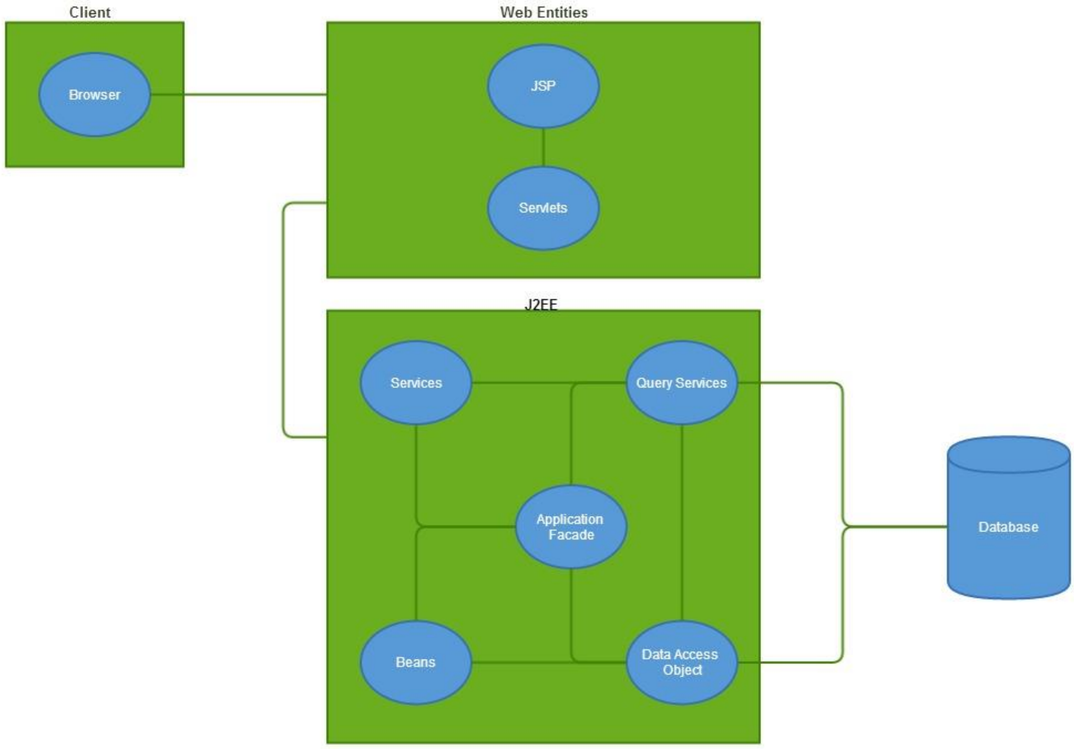

Figure 6: System architecture.

Figure 6 illustrates the technical architecture structuring the web application. The primary component is the J2EE component, which is Java Enterprise Edition. "*The Java EE platform provides an API and runtime environment for developing and running large-scale, multi-tiered, scalable, reliable, and secure network applications.*"[4]. in figure 6 the multi-tiered structure is illustrated displaying the division of web components, J2EE components and the database. The full system scope is divided into different tiers or levels. The first tier of the web application is the 'web components', these consist of the Java servlet pages, the Java servlets. This division will allow a contained implementation of the Model View Controller (MVC) design pattern of the web components. The web components tier is the only means of the client requesting information from the system and is the only means of receiving a response. The database is hosted on a MySQL server and is connected using a MySQL Java Database Connector (JDBC). The Interactions between these components is further document through the discussion of the current progress later in this chapter.

The application is being implemented using Java developed on Java Enterprise Edition Eclipse SDK. The application connects to a MySQL server. The MySQL workbench IDE has been used for design and development of the SQL statements forming the database schema. MySQL Work Bench also includes a modelling tool which has been used to reverse engineer the schema script to confirm the model meets the conceptual idea.

Xampp has been used for hosting the MySQL server and the application locally. Xampp is a cross platform web server package consisting of Apache, Tomcat, PHP and MySQL services.

The Drools workbench can be installed as a plugin for Eclipse enabling immediate project association within the Eclipse work space. Drools knowledge and rule files are designed and developed then as part of the Java application.

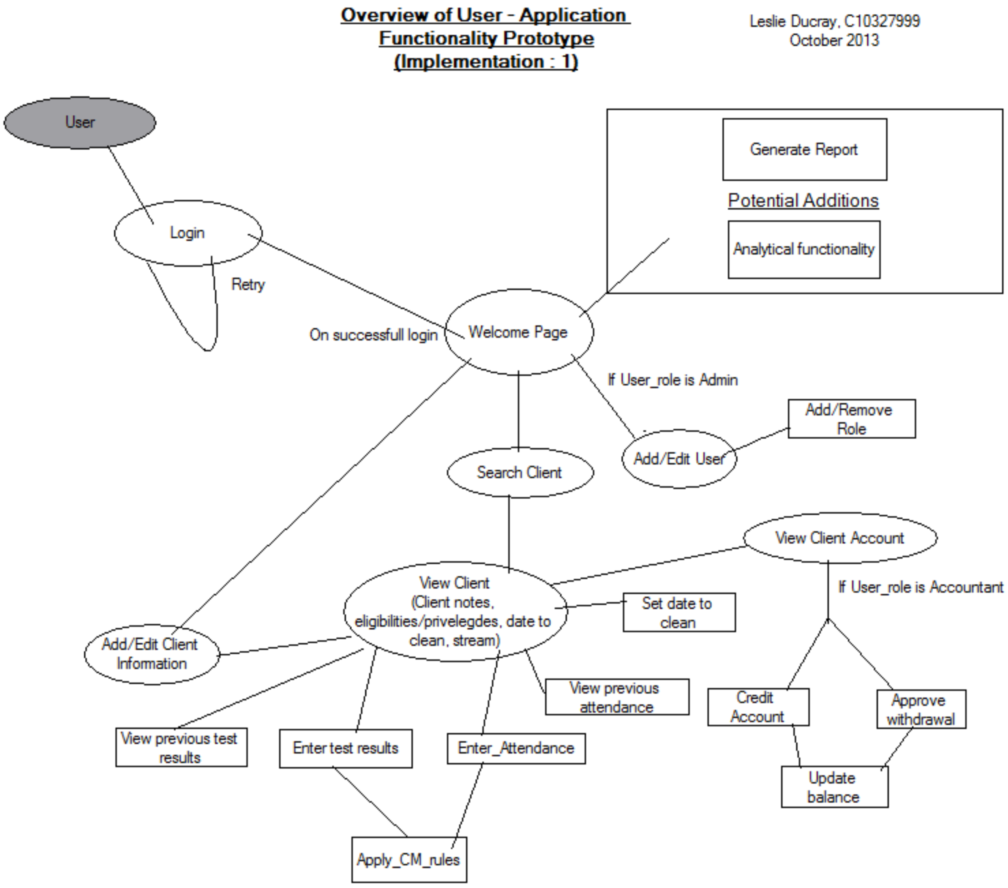

Figure 7: High level flow of the application front end.

Current Progress

The First Sprint involved the planning and development of ‘Login’ functionality. This begun by firstly by setting up the development environment and integrating the various technologies. The database schema had previously been implemented. A connection is made to the database using the JDBC and the database connection information is stored in a ‘context.xml’ file within the application files. This information includes the database name, its IP address and port, the username, password and the JDBC required. The data source information is stored as a bean in the ‘context.xml’ file, other beans are also stored in this file (17). This abstraction of database information is achieved using the Spring Framework. The Spring framework uses the JDBC to establish and manage a connection to the data source and to pass statements and the returned record sets across the connection. As the system is dependent on Java servlets and java beans, each instance of both needs to be declared and stored as XML in the ‘web.xml’ file. This XML structured document acts as a deployment descriptor; it determines how URLs will be mapped to the servlets (18).

Now that the servlet required for the login has been declared, it can now be defined. The ‘UserServlet.java’ will be requested for all actions involving queries, updates or processes involving user information. Using J2EE through Eclipse offers template files for Servlets, JavaBeans and JSP. Initializing a new servlet page declares the template of typical requirements such as the methods ‘doGet’ and ‘doPost’ which accept the HTTP request and response as parameters. As the Servlets will be required to process multiple actions, a list of valid actions are declared using Java enum class type which serves as a specific defined data constants. The list of valid actions will be actions required from the Java servlets pages (JSP). The first sprint will require a login form from a JSP to be submitted and ‘posted’ to the ‘UserServlet’ with the action of ‘Login’ which is a defined enum item. Within the servlet code is a ‘switch’ statement to catch the valid actions and execute the required code/action for each.

A Façade Interface can be used in an application as a ‘remote’ access layer to prevent revealing business components of the application to the client (19). They encapsulate the components of the application’s business model, which the clients can then access. A ‘UserFacade’ class is implemented with a method ‘authenticateUser’ which takes the ‘username’ and ‘password’ fields of the form submitted from the Login page. The ‘AuthenticateUser’ method is called from the servlet with the applicable parameters passed. The implementation of the façade method progressed by declaring a data Access Object (DAO), ‘UserDao’. A data access object is an attempt of creating an intermediate layer between the application and the database, whereby information from the data source is stored in an object of the application (19). Using Java beans designed to store the values of the retrieved attributes from the data source are an implementation of a DAO. Using a DAO keeps the data source hidden from the client. The application can then access and manipulate the values of the Java bean. Two Java beans are used for the login process, ‘UserBean’ and ‘UserRoleBean’. Firstly a query is made to the database to validate whether the logging in user exists and that the password matches. If the user is accepted, the ‘UserBean’ is populated with the user information from the database and a second query is performed to request what roles the particular user has, these values then populate a list of ‘UserRoleBean’s which is an attribute of the ‘UserBean’ class.

Java servlets can import various functionalities for dealing with HTTP requests, responses and sessions. If the user’s details are valid, the http session storage structure is populated by the user’s details and role permissions which will be used throughout the application for access and security. The ‘UserServlet’ then directs and dispatches the browser to a login confirmation page.

The second sprint implemented is to search for service users of the system. Currently the implementation is a basic search function to return any service uses fully or partially matching the service user’s name entered in by the user. In this instance we have ‘ServiceUserServlet’ which instantiates a list of ‘ServiceUserBeans’. These Beans are populated by the returned record set from a query made to the database. The Servlet initiates the method ‘SearchServiceUsers’ of the class ‘cmsQueryServiceUser’, a class which will be used for all general database queries in relation to service users.
The list of the service users returned is set to a variable of the JSP where it can be viewed by the user (system client).

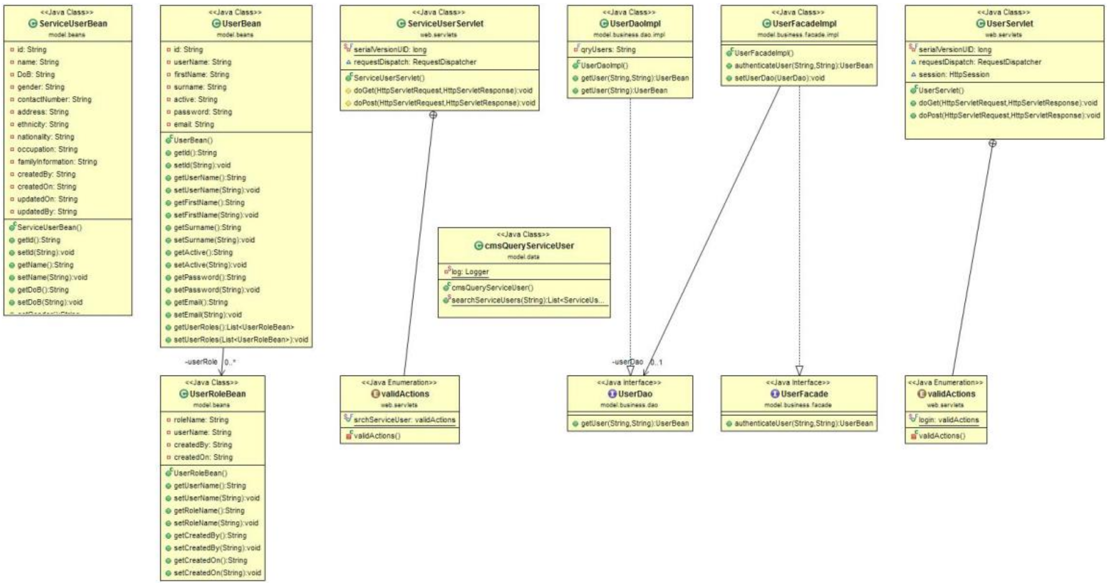

Figure 8: Class Diagram of current progress

Chapter 5 Challenges, future Work & Project Plan

Challenges

The largest foreseen challenge in the future of the application will be implementing and correctly evaluating the Contingency Management rules within the system and applying them to the application. Flow charts and decision diagrams displaying the reward criteria have been designed to aid this development process and the Drools language and model are being studied.

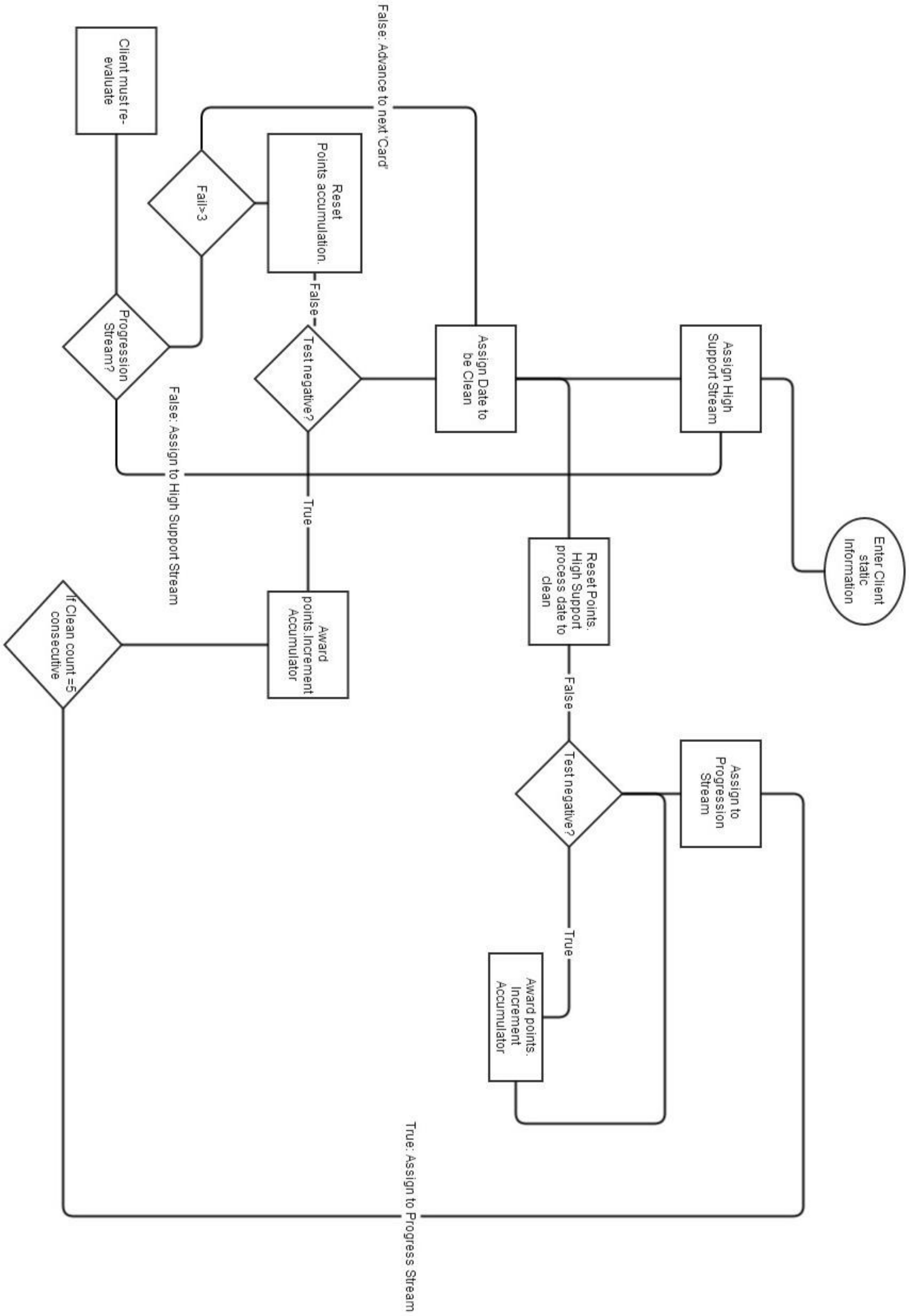
Client must re-evaluate
False: Advance to next 'Card'
Reset Points accumulation.
Fail>3
Progression Stream?
False
Test negative?
Assign Date to be Clean
Assign High Support Stream
False: Assign to High Support Stream
True
Enter Client static Information
Reset Points. High Support process date to clean
If Clean count =5 consecutive
Award points.Increment Accumulator
False
Test negative?
Assign to Progression Stream
True
Award points. Increment Accumulator
True: Assign to Progress Stream

Figure 9: Contingency Management process for rewarding/resetting service user's tested samples for substance abuse

In the coming months, priority is placed on the continual iteration of the devised system sprints to continue developing the system modules and ensuring they are thoroughly tested and meet the required specifications.

Testing

Primary testing will take place during each sprint iteration as with the process of the Agile design methodology. Following pre-written test cases, each aspect of the system will be thoroughly tested based on expected outcomes and intended design. These test cases will be passed or failed depending on the systems outputs. Each Implementation will undergo a full UAT (User Acceptability testing) ensuring the required system requirements are met and correctly implemented. Testing will be performed on the front end based on expected flow/directing of pages, form fields displayed, appearance, validation and client side functionality. The most crucial aspect of the system is the calculation and application of the Contingency Management rule criteria. These calculations will be performed by a following a pre-calculated route of actions for various permutations of the applicable scenarios and ensuring the system outputs match that of the pre-calculated test case.

Evaluation

Evaluation will be primarily based on the accuracy and correctness of the Contingency Management rules applied to the events of the system and how they improve the users experience and productivity in helping treat service users of the system. It will take time for solid feedback and evaluations of this system as is with the nature of the Contingency Management system. If it improves the work load, productivity and attitude towards work of the staff of a treatment facility and provides accurate and reliable results, this implementation may be considered a success.

Future Plans

December: To continue systematically designing, developing and testing independent modules of the system, and with each iteration incorporating them into the growing development. It is expected to have the majority of the system developed by the end of the year. This will include all non-rule based system functionality (e.g. Interactions with database, queries, inserts, updates, form processes, responses to client requests). Documenting all work as progression occurs.

January: Implementation of the rule based system using Drools and applying it to the functioning system. This process will be documented and explained in the final report. The Contingency Management rule base will be applied and associated with the existing Java application classes. Rules of this knowledge base will be called on the update of each Contingency Management applicable change in the system.

February: Full system tests and user accessibility testing. Evaluate with real users of a Contingency Management paper system, gain and document feedback. Document preliminary evaluations of the system.

March: Will be focussed on polishing off and improving the front end user interface, purely design and layout focussing on the heuristics and human computer interactions (HCI) properties.

The final report will take priority at this stage of the development life cycle and any remaining conclusions of the application. Ensuring the web application is functioning and deployed, ready for the Projects Fair in April.

Bibliography

1. Cummins B. ADRU press release story.: Health Research Board; 2013 [cited 2013 December 3. Available from: http://www.hrb.ie/health-information-in-house-research/alcohol-drugs/adru-news/adru-press-release-story/release/169/.
2. Drug Treatment Centre Board Annual Report 2011. ; 2011.
3. Geoghegan T. Merchants Quay Ireland Homeless and Drug Services Annual Report 2012. ; 2012.
4. Pilling S, Strang J, Gerada C. Psychosocial interventions and opioid detoxification for drug misuse: summary of NICE guidance. [Online].; 2007 [cited 2013 December 3. Available from: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1934496/.
5. Higgins ST, Petry NM. Contingency Management: Incentives for Sobriety. 1999; 23(2): p. 122-127.
6. Higgins ST, Silverman K, Heil SH. Contingency Management in Substance Abuse Treatment: Guilford Press; 2007.
7. Petry NM, Martin B, Cooney JL, Kranzler HR. Give them prizes and they will come: Contingency management for treatment of alcohol dependence. Journal of Consulting and Clinical Psychology, Vol 68(2). 2000;: p. 334.
8. Petry NM. Contingency Management for Substance Abuse Treatment: A Guide to Implementing This Evidence-Based Practice: Routledge; 2013.
9. Moos RH. Theory-based active ingredients of effective treatments for substance use disorders. Drug and Alcohol Dependence. 2007 May; 88(2-3).
10. Start Developing iOS Apps Today. [Online].; 2013 [cited 2013 November 29. Available from: https://developer.apple.com/library/ios/referencelibrary/GettingStarted/RoadMapiOS/index.html.
11. The MongoDB 2.4 Manual. [Online]. [cited 2013 December 3. Available from: http://docs.mongodb.org/manual/.
12. NET Framework System Requirements. [Online]. [cited 2013 December 3. Available from: http://msdn.microsoft.com/en-us/library/8z6watww(v=vs.110).aspx.
13. Evans I. Your First Cup: An Introduction to the Java EE Platform. [Online]. [cited 2013 October 10. Available from: http://docs.oracle.com/javaee/7/firstcup/doc/home.htm.
14. Johnson R, Hoeller J, Arendsen A, Sampaleanu C, Harrop R, Risberg T, et al. Spring - Java/J2EE Application Framework Reference Document. [Online]. [cited 2013 December 3. Available from: http://docs.spring.io/spring-framework/docs/1.2.8/reference/introduction.html.
15. team TJD. Drools Expert User Guide. [Online]. [cited 2013 October 10. Available from: http://docs.jboss.org/drools/release/5.2.0.Final/drools-expert-docs/html/.
16. Ambler S. Iteration Modeling: An Agile Best Practice. [Online]. [cited 2013 December 3. Available from: http://www.agilemodeling.com/essays/iterationModeling.htm.
17. The Apache Tomcat 5.5 Servlet/JSP Container. [Online]. [cited 2013 November 9. Available from: http://tomcat.apache.org/tomcat-5.5-doc/index.html.
18. The Deployment Descriptor: web.xml. [Online].; 2013 [cited 2013 November 9.

. Available from: https://developers.google.com/appengine/docs/java/config/webxml.

19 Alur D. Core J2EE Patterns: Best Practices and Design Strategies: Prentice Hall
. Professional; 2003.