Electronic mobile Commerce:

M- commerce or mobile e-commerce is a term for any type of business, or commercial transaction, that involves the transfer of information across the Internet. It covers a range of different types of businesses, from consumer based retail stores, through auction or music sites, to business exchanges trading goods and services between corporations. It is currently one of the most important aspects of the Internet to emerge.

Unlike traditional commerce that is carried out physically with effort of a person to go & get products, M-commerce has made it easier for human to reduce physical work and to save time. M-Commerce which was started no so long ago has taken a great leap in the world of technology, but the fact that has hindered the growth of m-commerce and e-commerce in general is security. Security is the challenge facing m-commerce today & there is still a lot of advancement made in the field of security.

The main advantage of m-commerce over traditional commerce is the user can browse online shops, compare prices and order merchandise from anywhere and at any time.

Shopping cart:

A shopping cart is a piece of software that acts as an online store's catalog and ordering process. Typically, a shopping cart is the interface between a company's online store and its deeper infrastructure. Contrary to popular belief among merchants, an online shopping cart does not process credit card payments. Instead, shopping cart software serves three other main purposes. First, it allows merchants to setup and manage online sales processes such as adding products, inventory management, order fulfillment and customer data collection. Second, it acts as the ordering interface for the customer allowing him/her to add/remove products (items) from the shopping cart and checkout. And lastly, it communicates the payment information to the payment gateway which encrypts the data and sends it to the banks for authorization if applicable. Shopping carts come in two forms: (1) Self-Hosted Software – This type of shopping cart is a dedicated piece of software that is hosted on the retailer’s own server. There are several varieties of licensed carts that range from open source free carts to proprietary packages that can cost thousands of dollars. (2) 3rd Party Hosted Software – Also known as an Application Service Provider (ASP), this type is shopping cart is hosted on the ASP’s secured servers and integrated with the retailers website using code snippets, JavaScript and/or API connections . Merchants who choose to host their own shopping carts are also responsible for securing their server and software against data theft which is a costly ongoing task. Generally, self-hosted shopping carts are only advisable for mid-to-large online retailers that have someone dedicated to maintaining the security of the online store.

Advantages of M-Commerce:

M-commerce obviously has numerous advantages over traditional commerce, the number one advantage being convenience of use. The user can browse online stores, compare prices and order merchandise sitting at home or anywhere they like. According to Forrester, 37 percent of online consumers use customer service more from Web retailers than traditional retailers because of its potential for ease of use and quick response times. "Once excellent proactive customer service is in place, merchants must build their brands around the promise of a satisfying experience," said Forrester's Christopher Kelley. "This means not only advertising a call center but also bragging about speedy response times and knowledgeable service reps". Dell, for example has adopted the same approach to selling their products both online and offline. This however, has proven beneficial for the company, since it helped them cut down on warehousing costs. One way in which the company has encouraged online ordering is by offering rebates on the products that are bought online. Other companies should follow Dell's example if they want to succeed in the online world.

Present Challenges Facing E-commerce and M-Commerce:

Speaking of obstacles, there are a lot of them that need to be uprooted before m-commerce can compete with traditional commerce. The biggest obstacle in the course of advancement of m-commerce is that the consumer's senses are limited to seeing and hearing the product. The second largest problem that e-commerce has been facing over the past few years is that of security. Traditional buyers and sellers are still paranoid about conducting business online. According to Hal Loevy, vice president of Global Marketing and Partnerships for SGSonSITE, "Despite all the noise about e-commerce, which is significant, companies still have to keep their old business practices: Can I trust who I am buying from? Who am I doing business with? What is their trading history? Am I obeying the law? Will I receive the goods as specified on screen and who do I approach if I have a problem?” According to emarketer.com, "70% of US consumers are concerned about online security; this discourages consumers from using credit cards to shop online (Payment One)".

Proposed project:

Kites M-commerce shopping cart, is a software Developed using state of the art technologies, based on various component, utilizing advanced programming concepts and techniques such as (RESTful web services, JPA, EJB, …) built on top of Java EE, Java SE and android platforms. Extreme programming is used throughout the development of the project.

Agile software development:

Agile software development is a group of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development, early delivery, continuous improvement and encourages rapid and flexible response to change. It is a conceptual framework that focuses on delivering working software with the minimum amount of work.

Agile principles:

The Agile Manifesto is based on twelve principles:

* Customer satisfaction by rapid delivery of useful software
* Welcome changing requirements, even late in development
* Working software is delivered frequently (weeks rather than months)
* Close, daily cooperation between business people and developers
* Projects are built around motivated individuals, who should be trusted
* Face-to-face conversation is the best form of communication (co-location)
* Working software is the principal measure of progress
* Sustainable development, able to maintain a constant pace
* Continuous attention to technical excellence and good design
* Simplicity the art of maximizing the amount of work not done is essential
* Self-organizing teams
* Regular adaptation to changing circumstances

Agile methods:

Well-known agile software development methods and/or process frameworks include:

* Adaptive Software Development (ASD)
* Agile Modeling
* Agile Unified Process (AUP)
* Crystal Methods (Crystal Clear)
* Disciplined Agile Delivery
* Dynamic Systems Development Method (DSDM)
* Extreme Programming (XP)
* Feature Driven Development (FDD)
* Lean software development
* Kanban (development)
* Scrum
* Scrum-ban

Agile practices:

Agile development is supported by a bundle of concrete practices suggested by the agile methods, covering areas like requirements, design, modeling, coding, testing, project management, process, quality, etc. Some notable agile practices include:

* Acceptance test-driven development (ATDD)
* Agile Modeling
* Backlogs (Product and Sprint)
* Behavior-driven development (BDD)
* Cross-functional team
* Continuous integration (CI)
* Domain-driven design (DDD)
* Information radiators (Scrum board, Kanban board, Task board, Burndown chart)
* Iterative and incremental development (IID)
* Pair programming
* Planning poker
* Refactoring
* Scrum meetings (Sprint planning, Daily scrum, Sprint review and retrospective)
* Test-driven development (TDD)
* Agile testing
* Time boxing
* Use case
* User story
* Story-driven modeling
* Velocity tracking

Extreme programming (XP):

Extreme programming (XP) is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements. As a type of agile software development, it advocates frequent "releases" in short development cycles, which is intended to improve productivity and introduce checkpoints at which new customer requirements can be adopted.

Other elements of extreme programming include: programming in pairs or doing extensive code review, unit testing of all code, avoiding programming of features until they are actually needed, a flat management structure, simplicity and clarity in code, expecting changes in the customer's requirements as time passes and the problem is better understood, and frequent communication with the customer and among programmers. The methodology takes its name from the idea that the beneficial elements of traditional software engineering practices are taken to "extreme" levels. As an example, Code reviews are considered a beneficial practice; taken to the extreme, code can be reviewed continuously, i.e. the practice of Pair programming.

XP attempts to reduce the cost of changes in requirements by having multiple short development cycles, rather than a long one. In this doctrine, changes are a natural, inescapable and desirable aspect of software-development projects, and should be planned for, instead of attempting to define a stable set of requirements.

Extreme programming also introduces a number of basic values, principles and practices on top of the agile programming framework.

XP Activities:

XP describes four basic activities that are performed within the software development process: coding, testing, listening, and designing. Each of those activities is described below.

* Coding

The advocates of XP argue that the only truly important product of the system development process is code – software instructions that a computer can interpret. Without code, there is no working product.

Coding can also be used to figure out the most suitable solution. Coding can also help to communicate thoughts about programming problems. A programmer dealing with a complex programming problem, or finding it hard to explain the solution to fellow programmers, might code it in a simplified manner and use the code to demonstrate what he or she means. Code, say the proponents of this position, is always clear and concise and cannot be interpreted in more than one way. Other programmers can give feedback on this code by also coding their thoughts.

* Testing

Extreme programming's approach is that if a little testing can eliminate a few flaws, a lot of testing can eliminate many more flaws.

Unit tests determine whether a given feature works as intended. A programmer writes as many automated tests as they can think of that might "break" the code; if all tests run successfully, then the coding is complete. Every piece of code that is written is tested before moving on to the next feature.

Acceptance tests verify that the requirements as understood by the programmers satisfy the customer's actual requirements.

System-wide integration testing was encouraged, initially, as a daily end-of-day activity, for early detection of incompatible interfaces, to reconnect before the separate sections diverged widely from coherent functionality. However, system-wide integration testing has been reduced, to weekly, or less often, depending on the stability of the overall interfaces in the system.

* Listening

Programmers must listen to what the customers need the system to do, what "business logic" is needed. They must understand these needs well enough to give the customer feedback about the technical aspects of how the problem might be solved, or cannot be solved. Communication between the customer and programmer is further addressed in the planning game.

* Designing

From the point of view of simplicity, of course one could say that system development doesn't need more than coding, testing and listening. If those activities are performed well, the result should always be a system that works. In practice, this will not work. One can come a long way without designing but at a given time one will get stuck. The system becomes too complex and the dependencies within the system cease to be clear. One can avoid this by creating a design structure that organizes the logic in the system. Good design will avoid lots of dependencies within a system; this means that changing one part of the system will not affect other parts of the system.

XP Practices:

Extreme programming has been described as having 12 practices, grouped into four areas:

* Fine-scale feedback
  + Pair programming
  + Planning game
  + Test-driven development
  + Whole team
* Continuous process
  + Continuous integration
  + Refactoring or design improvement
  + Small releases
* Shared understanding
  + Coding standards
  + Collective code ownership
  + Simple design
  + System metaphor
* Programmer welfare
  + Sustainable pace

Component-based software engineering (CBSE):

CBSE also known as component-based development (CBD) is a branch of software engineering that emphasizes the separation of concerns in respect of the wide-ranging functionality available throughout a given software system. It is a reuse-based approach to defining, implementing and composing loosely coupled independent components into systems. This practice aims to bring about an equally wide-ranging degree of benefits in both the short-term and the long-term for the software itself and for organizations that sponsor such software.

A computer running several software components is often called an application server. This combination of application servers and software components is usually called distributed computing.

Enterprise JavaBeans, Java EE and RESTful Web services are some technologies that are built on the concept of CBSE which are used in this project.

Enterprise JavaBeans (EJB):

EJB is a managed, server-side component architecture for modular construction of enterprise applications that encapsulates the business logic of an application.

The EJB specification details how an application server provides the following responsibilities

* Transaction processing.
* Integration with the persistence services offered by the Java Persistence API (JPA).
* Concurrency control.
* Event-driven programming using Java Message Service and Java EE Connector Architecture.
* Asynchronous method invocation.
* Job scheduling.
* Naming and directory services (JNDI).
* Interprocess Communication using RMI-IIOP and Web services.
* Security (JCE and JAAS).
* Deployment of software components in an application server.

Java Platform, Enterprise Edition (Java EE):

Java EE is Oracle's enterprise Java computing platform. The platform provides an API and runtime environment for developing and running enterprise software, including network and web services, and other large-scale, multi-tiered, scalable, reliable, and secure network applications. Java EE extends the Java Platform, Standard Edition (Java SE) providing an API for object-relational mapping, distributed and multi-tier architectures, and web services. The platform incorporates a design based largely on modular components running on an application server. Software for Java EE is primarily developed in the Java programming language.

Web service:

A Web service is a method of communication between two electronic devices over a network. It is a software function provided at a network address over the web with the service always on.

The W3C defines a Web service generally as:

a software system designed to support interoperable machine-to-machine interaction over a network.

Representational state transfer (REST):

REST is an architectural style consisting of a coordinated set of architectural constraints applied to components, connectors, and data elements, within a distributed hypermedia system. REST ignores the details of component implementation and protocol syntax in order to focus on the roles of components, the constraints upon their interaction with other components, and their interpretation of significant data elements.

The REST architectural style is also applied to the development of web services as an alternative to other distributed-computing specifications such as SOAP. One can characterize web services as "RESTful" if they conform to the constraints described in the architectural constraints section.

HTTP based RESTful web service APIs are defined with these aspects

* Base URI, such as (http://example.com/resources/).
* An Internet media type for the data. This is often JSON but can be any other valid Internet media type (e.g. XML, Atom, microformats, images, etc.).
* Standard HTTP methods (e.g., GET, PUT, POST, or DELETE).
* Hypertext links to reference state.
* Hypertext links to reference related resources.

Object databases:

An object database (also object-oriented database management system (OODBMS)) is a database management system in which information is represented in the form of objects as used in object-oriented programming. Object databases are different from relational databases which are table-oriented. Object-relational databases are a hybrid of both approaches.

ObjectDB:

ObjectDB is an object database for Java. That support two standard Java APIs - (JPA and JDO).

Java Persistence API (JPA):

The JPA is a Java programming language application programming interface specification that describes the management of relational data in applications using Java Platform, Standard Edition and Java Platform, Enterprise Edition.