DETAILED LEVEL DESIGN DOCUMENT OF DIGITAL SIGNATURES FOR MOBILE USERS IN THE SERVER END

Version number	2
CREATED BY	ARULPRAKASAM RAVINTHIRAN
DATE OF CREATION	Dec 20, 2014
TEAM MEMBERS	1. Arulprakasam Ravinthiran 2. Bernado Macedo 3. John Merkowsky
SUPERVISOR	Professor Carlisle Adams
COMMENTS	Version 2 has the project split up into server and client. Project report discusses the overall working of the project. High level design document of the server contains the name of all the classes. This document discusses the methods in each class of the server.

Contents

Model: Tables

tbl_userdetails

Column name	Type	Constraints
UserID	int(11)	Primary key
Name	varchar (45)	Not Null
FamilyName	varchar (45)	
PrimaryMobileNumber	varchar (20)	Unique, Not null
SecondaryMobileNumber	varchar (20)	
PrimaryEmailID	varchar (60)	Unique, Not Null
SecondaryEmailID	varchar (60)	
HasCertificate	tinyint(1)	NOT NULL, Default 0
Country	varchar (45)	NOT NULL
Province	varchar (45)	NOT NULL
City	varchar (45)	NOT NULL
ZipCode	varchar (10)	NOT NULL
Street	varchar (45)	NOT NULL

tbl_documentdetails

Column name	Type	Constraints
TrustCode	int(11)	Primary key
InitiatorUserID	int(11)	Foreign key references **tbl_UserDetails**(UserID)
OrigFile	Medium Blob	NOT NULL
MIMEType	varchar(100)	

UploadedTime	Timestamp	NOT NULL, Default current_timestamp

tbl_digitalsignaturedetails

Column name	Type	Constraints
TrustCode	int(11)	Foreign key references **tbl_DocumentDetails** (TrustCode)
UserID	int(11)	Foreign key references **tbl_UserDetails** (UserID)
SignedFile	Blob	
SignedTime	Timestamp	DEFAULT CURRENT_TIMESTAMP
ReasonForSigning	varchar(45)	NOT NULL DEFAULT “Not Applicable”
HasSigned	Tinyint(1)	NOT NULL Default 0
IsValid	Tinyint(1)	
		Primary key (TrustCode, UserID)

tbl_certificatedetails

Column name	Type	Constraints
CertificateID	int(11)	Primary key
UserID	int(11)	Foreign key references **tbl_UserDetails** (UserID)
CertificateFile	Blob	NOT NULL

tbl_revokedCertificateDetails

Column name	Type	Constraints
RevocationID	int(11)	Primary key
CertificateID	int(11)	Foreign key references **tbl_CertificateDetails** (CertificateID)
ReasonForRevocation	Varchar(155)	NOT NULL
RevokedTime	Timestamp	NOT NULL

Model: Bean classes

1. UserBean

Class name

Public class UserBean

Functionality

Bean class that defines the user

Variable names

1. private UserID userid
2. private String name
3. private String familyName
4. private PhoneNumber primaryNumber
5. private PhoneNumber secondaryNumber
6. private Boolean hasCertificate
7. private Address address

Methods

Signature	Functionality
public UserId getUserId()	Getter method for User ID
public void setUserId(UserId userId)	Setter method for user id
public String getName()	Getter method for name.
public void setName(String name)	Setter method for name.
public String getFamilyName()	Getter method for family name.
public void setFamilyName(String familyName)	Setter method for family name.
public Address getAddress()	Getter method for address.
public void setAddress(Address address)	Setter method for address.
public PhoneNumber getPrimaryNumber()	Getter method for primary phone number.
public void setPrimaryNumber(PhoneNumber primaryNumber)	Setter method for primary phone number.
public PhoneNumber getSecondaryNumber()	Getter method for secondary phone number.
public void setSecondaryNumber(PhoneNumber secondaryNumber)	Setter method for secondary phone number.
public EmailAddress getPrimaryEmail()	Getter method for primary email address.
public void setPrimaryEmail(EmailAddress primaryEmail)	Setter method for primary email address.
public EmailAddress getSecondaryEmail()	Getter method for secondary email address.
public void setSecondaryEmail(EmailAddress secondaryEmail)	Setter method for secondary email address.
public boolean hasCertificate()	Getter method for certificate status.
public void setHasCertificate(boolean hasCertificate)	Setter method for certificate status.

2. CertificateBean

Class name

Public class CertificateBean

Functionality

Bean class that defines the certificate.

Variable names

1. private UserID userId
2. private CertificateId certificateId
3. private byte[] certificateFile

Methods

Signature	Functionality
public UserId getUserId()	Getter method for User ID
public void setUserId(UserId userId)	Setter method for user id
public CertificateId getCerificateId ()	Getter method for Certificate Id.
public void setCertificateId (CertificateId certificateId)	Setter method for Certificate Id.
public byte[] getCertificateFile()	Getter method for certificate file.
public void setCertificateFile(byte[] certificateFile)	Setter method for certificate file.

3. RevokedCertificateBean

Class name

Public class RevokedCertificateBean

Functionality

Bean class that defines the certificate.

Variable names

1. private RevocationID revocationId
2. private CertificateId certificateId
3. private String revokingReason
4. private Timestamp revokedTime

Methods

Signature	Functionality
public RevocationId getRevocationId()	Getter method for Revocation ID
public void setRevocationId(RevocationId revocationId)	Setter method for revocation id
public CertificateId getCerificateId ()	Getter method for Certificate Id.
public void setCertificateId (CertificateId certificateId)	Setter method for Certificate Id.

public String getRevokingReason()	Getter method for revoking reason.
public void setRevokingReason (String revokingReason)	Setter method for revoking reason.
public Timestamp getRevokedTime()	Getter method for revoked time.
public void setRevokedTime(Timestamp revokedTime)	Setter method for revoked time.

4. DocumentBean

Class name

Public class DocumentBean

Functionality

Bean class that defines the document.

Variable names

1. private UserID userId
2. private TrustCode trustCode
3. private MimeType mimeType
4. private byte[] documentFile
5. private Timestamp uploadedTime
6. private String fileName

Methods

Signature	Functionality
public UserId getUserId()	Getter method for User ID
public void setUserId(UserId userId)	Setter method for user id
public String getFileName ()	Getter method for file name.
public void setFileName(String fileName)	Setter method for file name.
public TrustCode getTrustCode()	Getter method for trust code.
public void TrustCode(TrustCode trustCode)	Setter method for trust code.
public MimeType getMimeType()	Getter method for mime type.
public void setMimeType(MimeType mimeType)	Setter method for mime type.
public byte[] getDocumentFile()	Getter method for document file.
public void setDocumentFile(byte[] documentFile)	Setter method for document file.
public Timestamp getUploadedTime()	Getter method for uploaded time.
public void setUploadedTime(Timestamp uploadedTime)	Setter method for uploaded time.

5. DigitalSignatureBean

Class name

Public class DigitalSignatureBean

Functionality

Bean class that defines the digital signature.

Variable names

5. private UserID userId
6. private TrustCode trustCode
7. private String reasonForSigning
8. private byte[] signedFile
9. private Timestamp signedTime
10. private boolean hasSigned
11. private boolean isValid

Methods

Signature	Functionality
public UserId getUserId()	Getter method for User ID
public void setUserId(UserId userId)	Setter method for user id
public String getFileName ()	Getter method for file name.
public void setFileName(String fileName)	Setter method for file name.
public TrustCode getTrustCode()	Getter method for trust code.
public void TrustCode(TrustCode trustCode)	Setter method for trust code.
public String getSigningReason()	Getter method for reason for signing.
public void setSigningReason(String signingReason)	Setter method for reason for signing.
public byte[] getSignedFile()	Getter method for signed file.
public void setSignedFile(byte[] signedFile)	Setter method for signed file.
public Timestamp getSignedTime()	Getter method for signed time.
public void setSignedTime(Timestamp signedTime)	Setter method for signed time.
public boolean hasSigned()	Getter method for has signed boolean.
public void setHasSigned(boolean hasSigned)	Setter method for has signed boolean.
public boolean isValid()	Getter method for isValid boolean.
public void setIsValid(boolean isValid)	Setter method for isValid boolean.

Model: Datatypes

1. Address

Class name

Public class Address

Functionality

Data type that defines the address

Variable names

```
private String street
private String city
private String state
private String country
private String zip
```

Constructor

public Address(String country, String state, String city, String zip, String street) throws InvalidInputException

Methods

Signature	Functionality
public String getStreet()	Getter method for street
public String getCity()	Getter method for city
public String getState()	Getter method for state
public String getCountry()	Getter method for country
public String getZip()	Getter method for zip
public String getAddress()	Getter method for address
private void setAddress(String street, String city, String state, String country, String zip) throws InvalidInputException	Setter method for address
private boolean isValid()	Method for validation of address
public String toString()	Overridden method of the default 'to string' method

2. CertificateId

Class name

Public class CertificateId

Functionality

Data type that defines the certificate id.

Variable names

private int id

Constructor

public CertificateId(int id) throws InvalidInputException

Methods

Signature	Functionality
public int getId()	Getter method for certificate id
private void setId(int id) throws InvalidInputException	Setter method for certificate id
private boolean isValid()	Method that validates the certificate id

3. EmailAddress

Class name

Public class EmailAddress

Functionality
This is a data type that defines email address which is an open source code by the author Les Hazlewood.

4. PhoneNumber

Class name
Public class PhoneNumber

Functionality
Data type that defines the phone number.

Variable names
private int area
private int exch
private int ext

Constructor
public PhoneNumber(int area, int exch, int ext) throws InvalidInputException
public PhoneNumber(String phoneNumber)throws InvalidInputException

Methods

Signature	Functionality
public String getPhoneNumber()	Getter method for phone number
private void setPhoneNumber(String phoneNumber) throws InvalidInputException	Setter method for phone number
private void setPhoneNumber(int area, int exch, int ext) throws InvalidInputException	Setter method for phone number
private boolean isValid()	Method that validates the phone number
private boolean isValid(String phone)	Method that validates the phone number
public boolean equals(Object y)	Method that validates phone number
public String toString()	Overridden method that converts a phone number to string
public int hashCode()	Method that satisfies the hash code contract

5. RevocationId

Class name
Public class RevocationId

Functionality
Data type that defines the revocation id.

Variable names
private int id

Constructor
public RevocationId(int id) throws InvalidInputException

Methods

Signature	Functionality
public int getId()	Getter method for revocation id
private void setId(int id) throws InvalidInputException	Setter method for revocation id
private boolean isValid()	Method that validates the revocation id

6. Row

Class name
Public class Row

Functionality
Data type that defines a row that is fetched for the end user.

Variable names
private UserBean user
private DigitalSignatureBean signature

Constructor
public Row(UserBean user, DigitalSignatureBean signature)

Methods

Signature	Functionality
public UserBean getUser()	Getter method for user bean
public DigitalSignatureBean getDigitalSignature()	Getter method for digital signature bean

7. SignatureFile

Class name
Public class SignatureFile

Functionality
Data type that defines a signature file

Variable names
private String sha256Hash

Constructor
public SignatureFile(Blob blob, String hash) throws InvalidInputException, SerialException, SQLException

public SignatureFile(byte[] bytes, String hash) throws SerialException, SQLException, InvalidInputException

Methods

Signature	Functionality
public String getHash()	Getter method for hashing algorithm used.
private void setHash(String hash) throws InvalidInputException	Setter method for hashing algorithm used.
private boolean isValid()	Method that validates the signature file

8. TrustCode

Class name

Public class TrustCode

Functionality

Data type that defines the trust code of a document.

Variable names

private int trustCode

Constructor

public TrustCode(int trustCode) throws InvalidInputException

Methods

Signature	Functionality
public int getTrustCode()	Getter method for trust code
private void setTrustCode (int TrustCode) throws InvalidInputException	Setter method for trust code.
private boolean isValid()	Method that validates the trust code.

9. UserId

Class name

Public class UserId

Functionality

Data type that defines the user id.

Variable names

private int id

Constructor

public UserId (int id) throws InvalidInputException

Methods

Signature	Functionality
public int getId()	Getter method for user id
private void setId (int id) throws	Setter method for user id.

InvalidInputException	
private boolean isValid()	Method that validates the user id.

Model: Business classes

1. UserBiz

Class name
Public class UserDetails

Functionality
Creates a user record in the database by interacting with the JDBC classes.

Variable names
private Persistence persistence

Constructor
public UserBiz() throws DatabaseException

Methods

Signature	Functionality
public UserId createUser(UserBean user) throws DatabaseException, InvalidInputException	Method that creates a user for the application.
public UserBean retrieveUser(EmailAddress email) throws DatabaseException	Method that retrieves a user given an email.
public UserBean retrieveUser(UserId userId) throws DatabaseException	Method that retrieves a user given a user id.
public void setCertificateStatus(UserId userId, boolean status) throws DatabaseException	Method that sets the certificate status of a particular user.

2. CertificateBiz

Class name
Public class CertificateCreation

Functionality
Creates a certificate record for the given user in the database by interacting with the JDBC classes.

Variable names
private Persistence persistence

Constructor
public CertificateBiz() throws DatabaseException

Methods

Signature	Functionality
public CertificateId createCertificate(CertificateBean certificate) throws DatabaseException	Method that creates a certificate.
public CertificateBean retrieveCertificate(UserId userId) throws DatabaseException	Method that retrieves a certificate.
public void revokeCertificate(CertificateId certificateId) throws DatabaseException	Method that revokes a certificate.

3. DocumentBiz

Class name

Public class DocumentDetailsBiz

Functionality

Creates a document record in the database by interacting with the JDBC classes.

Variable names

private Persistence persistence

Constructor

public DocumentBiz() throws DatabaseException

Methods

Signature	Functionality
public TrustCode createDocument(DocumentBean document) throws DatabaseException	Method that creates a document.
public DocumentBean retrieveDocument(TrustCode trustCode) throws DatabaseException	Method that retrieves a document.
public void removeDocument(TrustCode trustCode) throws DatabaseException	Method that removes a document from the database.

4. DigitalSignatureBiz

Class name

Public class DigitalSignaturesBiz

Functionality

Creates a digital signature record in the database by interacting with the JDBC classes.

Variable names

private Persistence persistence

Constructor

public DigitalSignatureBiz () throws DatabaseException

Methods

Signature	Functionality
public Map<Integer,Row> retrieveSignature(TrustCode trustCode) throws DatabaseException	Method that retrieves a digital signature.
public void insertSigner(DigitalSignatureBean signature) throws DatabaseException	Method that inserts a signature record.
public void updateSignatureParameters(UserId userid, TrustCode trCode, boolean hasSigned, boolean isValid, String signingReason, byte[] signedFile) throws DatabaseException	Method that updates the parameters of a digital signature.
public boolean validateCoSigner(TrustCode trCode, UserId userId) throws DatabaseException	Method that validates a co-signer.
public boolean validateSignature(UserId userId,DigitalSignatureBean digSigBn, DocumentBean docBn) throws CertificateException, IOException, DatabaseException	Method that validates a digital signature from the database.

5. SignatureManager

Class name

Public class SignatureManager

Functionality

Business class that verifies a digital signature.

Variable name

Provider BC

Constructor

public SignatureManager()

Methods

Signature	Functionality
public boolean verifySignature(byte[] cert, byte[] sigBytes, byte[] data)	Method that verifies a digital signature.
public boolean verifySignature(PublicKey publicKey, byte[] sigBytes, byte[] data)	Method that verifies a digital signature.
public boolean verifySignature(X509Certificate certificate, byte[] sigBytes, byte[] data)	Method that verifies a digital signature.

6. CertificateManager

Class name

Public class Certificatemanager

Functionality
Business class that creates and verifies a digital certificate.

Variable name
Provider BC

Constructor
public CertificateManager()

Methods

Signature	Functionality
public X509Certificate createRootCertificate()	Method that creates a root certificate.
public X509Certificate createUserCertificate (PublicKey userPublicKey, UserBean userBn)	Method that creates a user certificate.
public void printCertificate(X509Certificate cert, String filename)	Method that prints a digital certificate.
public boolean verifyUserCertificate (X509Certificate certificate)	Method that verifies a user certificate.

Model: Util classes

1. BCRSAPrivateCrtKey

Class name
Public class BCRSAPrivateCrtKey

Functionality
Util class that is for RSA private keys with certificate factors included.

Variable names
private BigInteger publicExponent
private BigInteger primeP
private BigInteger primeQ
private BigInteger primeExponentP
private BigInteger primeExponentQ
private BigInteger crtCoefficient

Constructors
1. `BCRSAPrivateCrtKey(RSAPrivateCrtKeyParameters key)`
2. `BCRSAPrivateCrtKey(RSAPrivateCrtKeySpec spec)`
3. `BCRSAPrivateCrtKey(RSAPrivateCrtKey key)`
4. `BCRSAPrivateCrtKey(PrivateKeyInfo info)`
5. `BCRSAPrivateCrtKey(RSAPrivateKey key)`

Methods

Signature	Functionality

public String getFormat()	Method that returns the encoding format.
public byte[] getEncoded()	Method that returns an encoded object.
public BigInteger getPublicExponent()	Method that returns the public key.
public BigInteger getPrimeP()	Method that returns the prime component P.
public BigInteger getPrimeQ()	Method that returns the prime component Q.
public BigInteger getPrimeExponentP()	Method that returns the prime exponent for P.
public BigInteger getPrimeExponentQ()	Method that returns the prime exponent for Q.
public BigInteger getCrtCoefficient()	Method that returns the certificate coefficient.
public int hashCode()	Method that returns the hash code.
public boolean equals(Object o)	Overridden method that compares the given object for RSA public and private key components.
public String toString()	Overridden method that gives a string.

2. BCRSAPrivateKey

Class name

Public class BCRSAPrivateKey

Functionality

Util class that is for RSA private keys.

Variable names

private static BigInteger ZERO = BigInteger.valueOf(0)
protected BigInteger modulus
protected BigInteger privateExponent
private transient PKCS12BagAttributeCarrierImpl attrCarrier

Constructors

```
1. protected BCRSAPrivateKey()
2. BCRSAPrivateKey(RSAKeyParameters key)
3. BCRSAPrivateKey(RSAPrivateKeySpec spec)
4. BCRSAPrivateKey(RSAPrivateKey key)
```

Methods

Signature	Functionality
public String getFormat()	Method that returns the encoding format.
public byte[] getEncoded()	Method that returns an encoded object.
public BigInteger getPrivateExponent()	Method that returns the private exponent.

public int hashCode()	Method that returns the hash code.
public boolean equals(Object o)	Overridden method that compares the given object for RSA public and private key components.
public String toString()	Overridden method that gives a string.
public String getAlgorithm ()	Method that returns the algorithm name.
public void setBagAttribute(ASN1ObjectIdentifier oid, ASN1Encodable attribute)	Method that sets the given attributes.
public ASN1Encodable getBagAttribute `(ASN1ObjectIdentifier oid)`	Method that returns the ANSI encodable attributes.
public Enumeration<?> getBagAttributeKeys()	Method that returns the required attributes.
private void readObject(ObjectInputStream in)	Method that reads the given objects.
private void writeObject(ObjectOutputStream out)	Method that writes the given object.

3. BCRSAPublicKey

Class name

Public class BCRSAPublicKey

Functionality

Util class that is for BC RSA publc keys.

Variable names

protected BigInteger modulus
protected BigInteger publicExponent
private transient AlgorithmIdentifier algorithmIdentifier

Constructors

```
1. BCRSAPublicKey (RSAKeyParameters key)
2. BCRSAPublicKey (RSAPublicKeySpec spec)
3. BCRSAPublicKey (RSAPublicKey key)
4. BCRSAPublicKey (SubjectPublicKeyInfo info)
```

Methods

Signature	Functionality
private void populateFromPublicKeyInfo (SubjectPublicKeyInfo info)	Method that sets the attributes from the given key info.
public String getFormat()	Method that returns the encoding format.
public byte[] getEncoded()	Method that returns an encoded object.
public BigInteger getPublicExponent()	Method that returns the public exponent.

public int hashCode()	Method that returns the hash code.
public boolean equals(Object o)	Overridden method that compares the given object for RSA public and private key components.
public String toString()	Overridden method that gives a string.
public String getAlgorithm ()	Method that returns the algorithm name.
public void setBagAttribute(ASN1ObjectIdentifier oid, ASN1Encodable attribute)	Method that sets the given attributes.
public ASN1Encodable getBagAttribute (ASN1ObjectIdentifier oid)	Method that returns the ANSI encodable attributes.
public Enumeration<?> getBagAttributeKeys()	Method that returns the required attributes.
private void readObject(ObjectInputStream in)	Method that reads the given objects.
private void writeObject(ObjectOutputStream out)	Method that writes the given object.

4. BigIntegerMath

Class name

Public class BigIntegerMath

Functionality

This class contains the methods that use various factorization and other algorithms.

Variable names

public static final BigInteger ZERO
public static final BigInteger ONE
public static final BigInteger TWO
public static final BigInteger THREE
public static final BigInteger FOUR

Methods

Signature	Functionality
public static BigInteger[] euclid(BigInteger a,BigInteger b) throws IllegalArgumentException	A non-recursive method of Euclid that returns an array of 3 BigIntegers.
public static BigInteger[] solveLinearDiophantine(BigInteger a, BigInteger b, BigInteger c) throws IllegalArgumentException	Method returns a particular solution (if any solutions exist) of linear equations of the form ax+by=c.
public static BigInteger lnr(BigInteger b, BigInteger m)	Method Computes the least nonnegative residue of b mod m, where m>0.
public static BigInteger[]	Returns a solution of x for linear congruences of the form ax

solveLinearCongruence(BigInteger a, BigInteger b, BigInteger m)	congruent to b (mod m)
public static double primeProbability(BigInteger n,int numPasses,SecureRandom sr)	Method implements the Rabin-Miller test.
public static BigInteger[] solveCRT(BigInteger[] residue, BigInteger[] modulus)	Method Finds simultaneous solutions to a linear system of congruences involving only one variable and multiple moduli.
public static BigInteger[] solveQuadratic(BigInteger a, BigInteger b, BigInteger c, BigInteger p, BigInteger q, int primeTolerance)	Method solves quadratic congruences ax^2+bx+c congruent to 0 mod n=pq.
public static BigInteger monteCarloFactor(BigInteger n,int maxArraySize) throws IllegalArgumentException	Monte Carlo factorization method returns a Monte Carlo factor.
public static BigInteger pMinusOneFactor(BigInteger n) throws IllegalArgumentException	Pollard p-1 factorization-runs until a factor is found.
public static BigInteger sqrt(BigInteger m)	Method that finds a square root.
public static BigInteger logExhaustiveSearch(BigInteger base, BigInteger residue, BigInteger modulus)	This algorithm solves base^x = residue (mod modulus) for x using exhaustive search.
public static BigInteger logBabyStepGiantStep(BigInteger base, BigInteger residue, BigInteger modulus)	This algorithm solves base^x = residue (mod modulus) for x using baby step giant step.

5.DigSigMobServerUtils

Class name

Public class DigSigMobServerUtils

Functionality

Util class that sends SMS, email and generates a public key given a mobile number.

Methods

Signature	Functionality
public String generatePublicKeyE (String prmryMobNo)	Method generates a part of a public key.
public void sendEmail (String recipientEmail, String message, String	Method sends email.

subject)	
public void sendMail (TrustCode trCode, EmailAddress email, EmailAddress cosigEmail, boolean isInitiator, boolean isSignValid) throws DatabaseException	Overridden method that sends email to more than one id.
public void sendSMS (String mobNo, String message)	Method sends SMS.

6. EmailProvider

Class name

Public class EmailProvider

Functionality

Util class for sending email.

Variable names

```
private static final String username
private static final String password
private static final String host
private static final String port
private String response
private boolean sent
```

Methods

Signature	Functionality
public String getResponse()	Method that returns the response.
public boolean isSent()	Method that returns the status of the sent email.
public void sendMail(String toEmail, String subject, String msg)	public void sendMail(String toEmail, String subject, String msg)
public void setSent(boolean sent)	public void setSent(boolean sent)

7. HttpConnection

Class name

Public class HttpConnection

Functionality

Util class for assisting email sending.

Variable names

```
Private final String USER_AGENT
private int responseCode
```

Methods

Signature	Functionality
public int getResponseCode()	Method that returns the response code.
public void sendGet(String url)	Method for HTTP get request.
public void sendPost(String url, String urlParameters)	Method for HTTP post request.

8. KeyStoreProvider

Class name

Public class KeyStoreProvider

Functionality

Util class for KeyStoreProvider.

Variable names

```
private static final String KEYSTORE_FILE
private static final String KEYSTORE_PWD
private static final String KEYSTORE_INSTANCE
private static final String KEYSTORE_ALIAS
```

Methods

Signature	Functionality
public static KeyStore loadKeyStore()	Method to load a KeyStore object.
public static X509Certificate retrieveRootCertificate()	Method that retrieve Certificate Authority root certificate.
public static PrivateKey retrieveRootPrivatekey()	Method that retrieve the private key of the Certificate Authority.
public static void store(X509Certificate certificate, PrivateKey key)	Method that stores a root certificate and a private key inside the KeyStore.

9. SMSProvider

Class name

Public class SMSProvider

Functionality

Util class for sending SMS.

Variable names

```
private static final String targetURL
private static final String password
private String response
```

Methods

Signature	Functionality
public String getResponse()	Method that returns the response.
public void sendSMS(String phoneNumber, String message)	Method that sends SMS.

Model: JDBC connection classes

1. Command

Class name

public abstract class Command

Functionality

This is the base class of the Command design pattern, which provides a hierarchy of different Commands to be executed by the Persistence tier. This is an abstract class and each subclass must implement its run method.

Variable names

```
protected boolean isQuery
protected PreparedStatement preparedStatement
protected ResultSet resultSet
protected String sqlCommand
```

Methods

Signature	Functionality
public PreparedStatement getCommand(Connection conn) throws DatabaseException	Method that returns the response.
public boolean isQuery()	Method that returns the isQuery boolean.

protected abstract PreparedStatement prepareCommand(Connection conn)	Abstract method that should be defined by the inheriting classes.
public abstract void run() throws DatabaseException, InvalidInputException, MimeTypeParseException	Method that is used by the inheriting children classes.
public void setCommand(String sqlCommand)	Method for setting the sql command.
public void setResultSet(ResultSet resultSet)	Method for setting the result set.

2. ConnectionDB

Class name

Public class ConnectionDB

Functionality

Class that connects to the mySql database.

Variable names

```
private static final String JDBC_DRIVER
private static final String DB_URL
private static final String USER
private static final String PASSWORD
```

Methods

Signature	Functionality
public static Connection getConnectionObject() throws SQLException	Method that returns the connection object.
public static String getDriverObject()	Method that returns the driver object.

3. Persistence

Class name

Public class Persistence

Functionality

Class controls all the database commands. It follows the Singleton pattern: can't be instantiated more than once.

Variable names

```
private static Persistence persistenceInstance
private Connection connObj
private boolean connected
```

Constructor

private Persistence()

Methods

Signature	Functionality
public static Persistence getInstance()	Method that returns the available instance.
public void connect() throws DatabaseException	Method that creates a new connection with the database.
public void disconnect() throws DatabaseException	Method that disconnects with the database.
public void run(Command cmd) throws DatabaseException	Method that runs the SQL command.

4. CmdCreateCertificate

Class name

Public class CmdCreateCertificate

Functionality

Interacts with the database for certificate creation.

Variable names

```
private CertificateBean cert
```

Constructor

public CmdCreateCertificate (CertificateBean cert)

Methods

Signature	Functionality
protected PreparedStatement prepareCommand(Connection conn) throws DatabaseException	Overridden method for certificate details insertion.
public void run() throws DatabaseException	Overridden method for certificate details insertion.

5. CmdCreateRevokedCertificate

Class name
Public class CmdCreateRevokedCertificate

Functionality
Interacts with the database for revoked certificate record insertion.

Variable names
private RevokedCertificateBean revkdCert

Constructor
public CmdCreateRevokedCertificate

Methods

Signature	Functionality
protected PreparedStatement prepareCommand(Connection conn) throws DatabaseException	Overridden method for revoked certificate details insertion.
public void run() throws DatabaseException	Overridden method for revoked certificate details insertion.

6. CmdRetrieveCertificate

Class name
Public class CmdRetrieveCertificate

Functionality
Interacts with the database for certificate details retrieval.

Variable names
```
private CertificateBean cert
private UserId userId
```

Constructor
public CmdRetrieveCertificate (UserId userId)

Methods

Signature	Functionality
protected PreparedStatement prepareCommand(Connection conn) throws DatabaseException	Overridden method for certificate details retrieval.
public void run() throws DatabaseException	Overridden method for certificate details retrieval.
public CertificateBean getCert()	Method that returns the retrieved certificate bean.

7. CmdRetrieveMaxCertId

Class name
Public class CmdRetrieveMaxCertId

Functionality
Interacts with the database for maximum certificate id retrieval.

Variable names
private CertificateId maxCertId

Constructor
public CmdRetrieveMaxCertId()

Methods

Signature	Functionality
protected PreparedStatement prepareCommand(Connection conn) throws DatabaseException	Overridden method for maximum certificate id retrieval.
public void run() throws DatabaseException	Overridden method for maximum certificate id retrieval.
public CertificateId getMaxCertificateId()	Method that returns the fetched CertificateId.

8. CmdRetrieveRevokedCertificate

Class name
Public class CmdRetrieveRevokedCertificate

Functionality
Interacts with the database for revoked certificate details retrieval.

Variable names
```
private RevokedCertificateBean revkdCert
private EmailAddress email
```

Constructor
public CmdRetrieveRevokedCertificate (EmailAddress email)

Methods

Signature	Functionality
protected PreparedStatement prepareCommand(Connection conn) throws DatabaseException	Overridden method for revoked certificate details retrieval.
public void run() throws DatabaseException	Overridden method for revoked certificate details retrieval.

public RevokedCertificateBean getRevkdCert()	Method that returns the fetched bean.

9. CmdRetrieveUser

Class name

Public class CmdRetrieveUser

Functionality

Interacts with the database for user details retrieval.

Variable names

```
private UserBean user
private EmailAddress email
private UserId userId
```

Constructor

```
public CmdRetrieveUser (EmailAddress email)
public CmdRetrieveUser (UserId userId)
```

Methods

Signature	Functionality
protected PreparedStatement prepareCommand(Connection conn) throws DatabaseException	Overridden method for revoked certificate details retrieval.
public void run() throws DatabaseException	Overridden method for revoked certificate details retrieval.
public UserBean getUser()	Method that returns the fetched bean.

10. CmdUpdateUserCertStatus

Class name

Public class CmdUpdateUserCertStatus

Functionality

Interacts with the database for certificate status updation.

Variable names

```
private UserBean user
private boolean status
```

Constructor

public CmdUpdateUserCertStatus(UserId id, boolean status)

Methods

Signature	Functionality
protected PreparedStatement	Overridden method for certificate details insertion.

prepareCommand(Connection conn) throws DatabaseException	
public void run() throws DatabaseException	Overridden method for certificate details insertion.

11. CmdCreateDigitalSignature

Class name

Public class CmdCreateDigitalSignature

Functionality

Interacts with the database for digital signature details insertion.

Variable names

```
private DigitalSignatureBean digSig
```

Constructor

public CmdCreateDigitalSignature (DigitalSignatureBean digSig)

Methods

Signature	Functionality
protected PreparedStatement prepareCommand(Connection conn) throws DatabaseException	Overridden method for certificate details insertion.
public void run() throws DatabaseException	Overridden method for certificate details insertion.

12. CmdRetrieveDigitalSignature

Class name

Public class CmdRetrieveDigitalSignature

Functionality

Interacts with the database for digital signature details retrieval.

Variable names

```
private DigitalSignatureBean digSigBnObj
private UserBean userBnObj
private Map<Integer, Row> rowMap
private Row rowObj
private TrustCode trCode
```

Constructor

public CmdRetrieveDigitalSignature (TrustCode trCode)

Methods

Signature	Functionality
protected PreparedStatement prepareCommand(Connection conn) throws DatabaseException	Overridden method for revoked certificate details retrieval.
public void run() throws DatabaseException	Overridden method for revoked certificate details retrieval.
public Map<Integer,Row> getMap ()	Method that returns the fetched hash map.

13. CmdUpdateSignatureParameters

Class name

Public class CmdUpdateSignatureParameters

Functionality

Interacts with the database for signature details updation.

Variable names

```
private TrustCode trCode
private byte[] signedFile
private UserId userId
private Sring signingReason
private boolean isValid
private boolean hasSigned
```

Constructor

public CmdUpdateSignatureParameters(UserId id, TrustCode trCode,boolean hasSigned, boolean isValid, String signingReason, byte[] signedFile)

Methods

Signature	Functionality
protected PreparedStatement prepareCommand(Connection conn) throws DatabaseException	Overridden method for certificate details insertion.
public void run() throws DatabaseException	Overridden method for certificate details insertion.

14. CmdValidateCoSigner

Class name

Public class CmdValidateCoSigner

Functionality

Interacts with the database for co-signer validation.

Variable names

```
private TrustCode trCode
private UserId userId
private int count
```

Constructor

public CmdValidateCoSigner(TrustCode trCode, UserId userId)

Methods

Signature	Functionality
protected PreparedStatement prepareCommand(Connection conn) throws DatabaseException	Overridden method for certificate details insertion.
public void run() throws DatabaseException	Overridden method for certificate details insertion.
public boolean validateCoSigner()	Method that returns the validation result.

15. CmdCreateDocument

Class name

Public class CmdCreateDocument

Functionality

Interacts with the database for document details insertion.

Variable names

```
private DocumentBean document
```

Constructor

public CmdCreateDocument (DocumentBean document)

Methods

Signature	Functionality
protected PreparedStatement prepareCommand(Connection conn) throws DatabaseException	Overridden method for user details insertion.
public void run() throws DatabaseException	Overridden method for user details insertion.

16. CmdRetrieveDocument

Class name

Public class CmdRetrieveDocument

Functionality
Interacts with the database for document details retrieval.

Variable names
```
private DocumentBean document
private EmailAddress email
```

Constructor
public CmdRetrieveDocument (EmailAddress email)

Methods

Signature	Functionality
protected PreparedStatement prepareCommand(Connection conn) throws DatabaseException	Overridden method for revoked certificate details retrieval.
public void run() throws DatabaseException	Overridden method for revoked certificate details retrieval.
public DocumentBean getDocument()	Method that returns the fetched bean.

17. CmdRetrieveMaxTrustCode

Class name
Public class CmdRetrieveMaxTrustCode

Functionality
Interacts with the database for maximum trust code retrieval.

Variable names
private TrustCode maxTrCode

Constructor
public CmdRetrieveMaxTrustCode()

Methods

Signature	Functionality
protected PreparedStatement prepareCommand(Connection conn) throws DatabaseException	Overridden method for certificate details retrieval.
public void run() throws DatabaseException	Overridden method for certificate details retrieval.
public TrustCode getMaxTrustCode()	Method that returns the fetched trust code.

18. CmdCreateUser

Class name
Public class CmdCreateUser

Functionality
Interacts with the database for user details insertion.

Variable names
```
private UserBean user
```

Constructor
public CmdCreateUser (UserBean user)

Methods

Signature	Functionality
protected PreparedStatement prepareCommand(Connection conn) throws DatabaseException	Overridden method for user details insertion.
public void run() throws DatabaseException	Overridden method for user details insertion.

19. CmdRetrieveMaxUserId

Class name
Public class CmdRetrieveMaxUserId

Functionality
Interacts with the database for maximum certificate id retrieval.

Variable names
private UserId maxUserId

Constructor
public CmdRetrieveMaxUserId()

Methods

Signature	Functionality
protected PreparedStatement prepareCommand(Connection conn) throws DatabaseException	Overridden method for certificate details retrieval.
public void run() throws DatabaseException	Overridden method for certificate details retrieval.
public UserId getUserId()	Method that returns the retrieved User Id.

20. CmdRetrieveUser

Class name
Public class CmdRetrieveUser

Functionality
Interacts with the database for revoked certificate details retrieval.

Variable names
```
private UserBean user
private UserId userId
private EmailAddress email
```

Constructor
public CmdRetrieveUser (EmailAddress email)

Methods

Signature	Functionality
protected PreparedStatement prepareCommand(Connection conn) throws DatabaseException	Overridden method for revoked certificate details retrieval.
public void run() throws DatabaseException	Overridden method for revoked certificate details retrieval.
public UserBean getUser()	Method that returns the fetched bean.

Model: Exceptions

1. DatabaseException

Class name
Public class DatabaseException

Functionality
A custom Java exception class.

Constructor
public DatabaseException(String message)

2. InvalidInputException

Class name
Public class InvalidInputException

Functionality
A custom Java exception class.

Constructor
public InvalidInputException (String message)

Controller: JAVA classes

1.TrustCodeVerification

Class name
Public class DigSigMobController

Functionality
Control class responsible for trust code verification.

Methods

Signature	Functionality
public void verifyTrustCode (TrustCode trCode)	Method that verifies the trust code.

2.SignatureController

Class name
Public class SignatureController

Functionality
Control class responsible for digital signatures.

Methods

Signature	Functionality
public static int createSignature(DigitalSignatureBean signature, boolean isInitiator, EmailAddress[] emails)	Method that creates the signature by interacting with appropriate business classes.
public static int validateCoSigner(DocumentBean document, EmailAddress email)	Method that validates the co-signers and returns the public key of the initiator as the status.
public static int validateSigners (EmailAddress[] emails)	Method that validates the signers and returns the public key of the initiator as the status.

3. UserController

Class name

Public class UserController

Functionality

Control class responsible for user registration.

Methods

Signature	Functionality
public static int registerUser (UserBean user)	Method that creates a user record and returns the user id on successful creation..

4. CertificateController

Class name

Public class CertificateController

Functionality

Control class responsible for digital certificate related activities.

Methods

Signature	Functionality
public static int createUserCertificate(PublicKey publicKey, EmailAddress email)	Method creates a new certificate for the user and adds to the server database.
public static int validateKeys (String keyP1, String keyP2, EmailAddress email)	Method validates the entered mobile and email keys and returns the user id if all keys are correct.

5. DocumentController

Class name

Public class DocumentController

Functionality

Control class responsible for document related activities.

Methods

Signature	Functionality
public static int fileUpload(DocumentBean document, EmailAddress email)	Method returns the trust code of the uploaded file or error message.
public static	Method returns the map returned from the database.

HashMap<Integer, Row> verifyTrustcode(TrustCode trustCode)	

Controller: Starter class

Server

Class name

public class Server

Functionality

Starter class.

Variable names

```
public static int uniqueId
private ArrayList<ClientThread> list
private int port
private boolean keepGoing
```

Constructor

public Server(int port)

Methods

Signature	Functionality
public void start()	Method for starting the server.
protected void stop()	Method for stopping the server.
public synchronized void remove(int id)	Method that removes the client connection.
private void display(String msg)	Method for displaying the message.
public static void main(String[] args)	Main method.