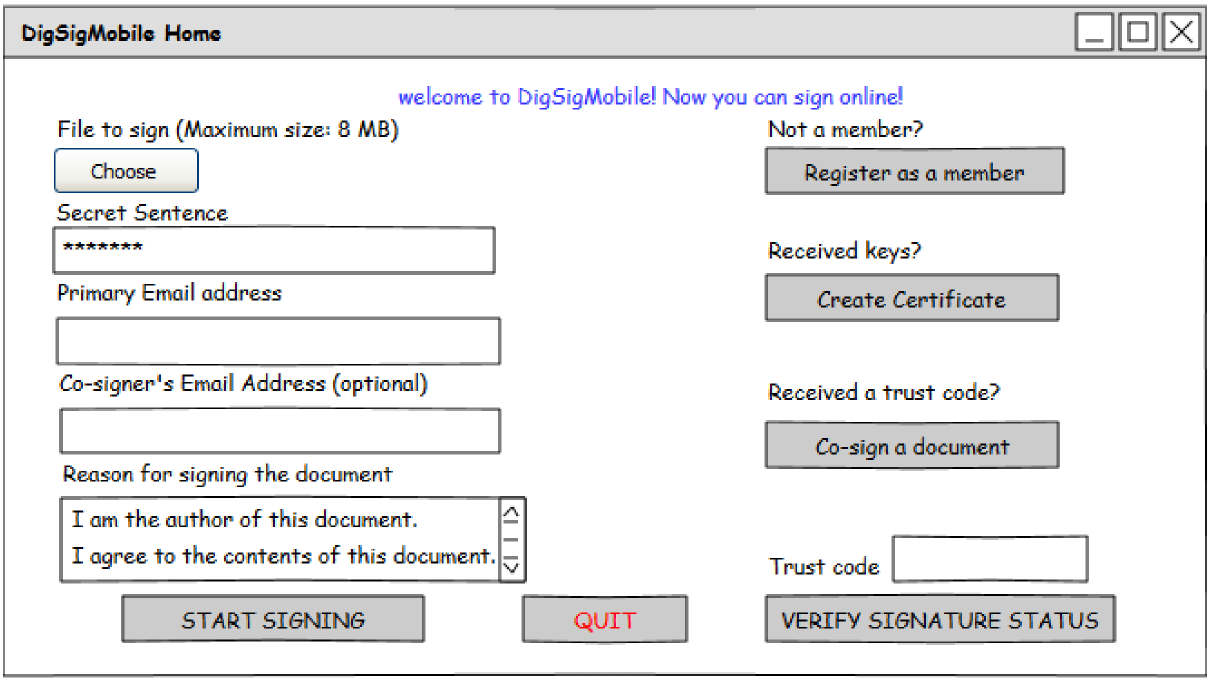
DETAILED LEVEL DESIGN DOCUMENT OF DIGITAL SIGNATURES FOR MOBILE USERS IN THE CLIENT END

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 client contains the name of all the classes. This document discusses the methods in each class of the client.

Contents

View: Screen shots and details of the desktop application

Home



welcome to DigSigMobile! Now you can sign online!

File to sign (Maximum size: 8 MB)

Secret Sentence: *******

Primary Email address

Co-signer's Email Address (optional)

Reason for signing the document:
I am the author of this document.
I agree to the contents of this document.

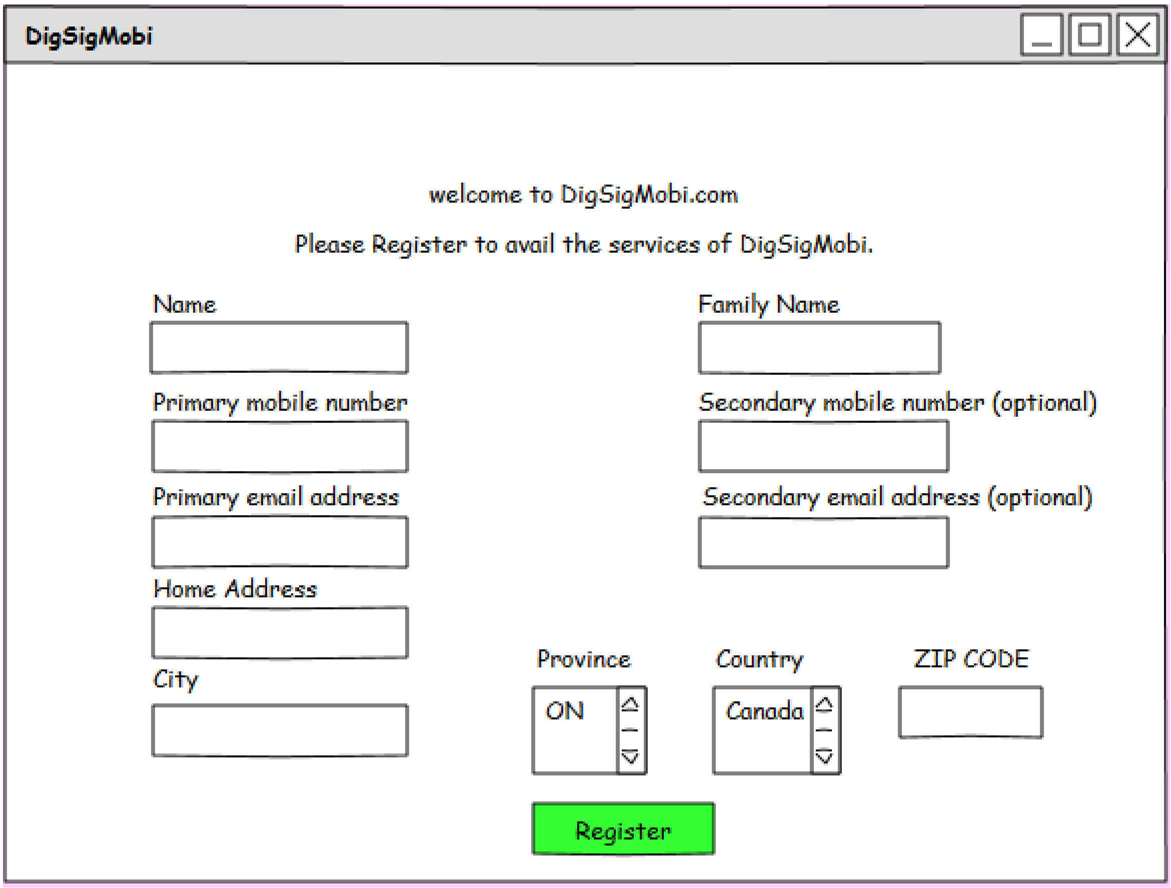
Not a member?

Received keys?

Received a trust code?

Trust code

Registration



welcome to DigSigMobi.com

Please Register to avail the services of DigSigMobi.

Name

Family Name

Primary mobile number

Secondary mobile number (optional)

Primary email address

Secondary email address (optional)

Home Address

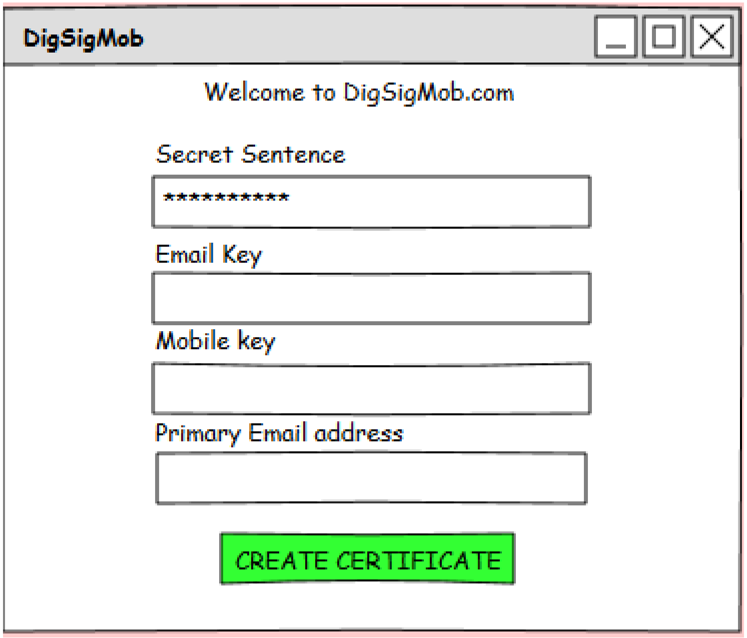
City

Province: ON

Country: Canada

ZIP CODE

Certificate Creation



DigSigMob

Welcome to DigSigMob.com

Secret Sentence

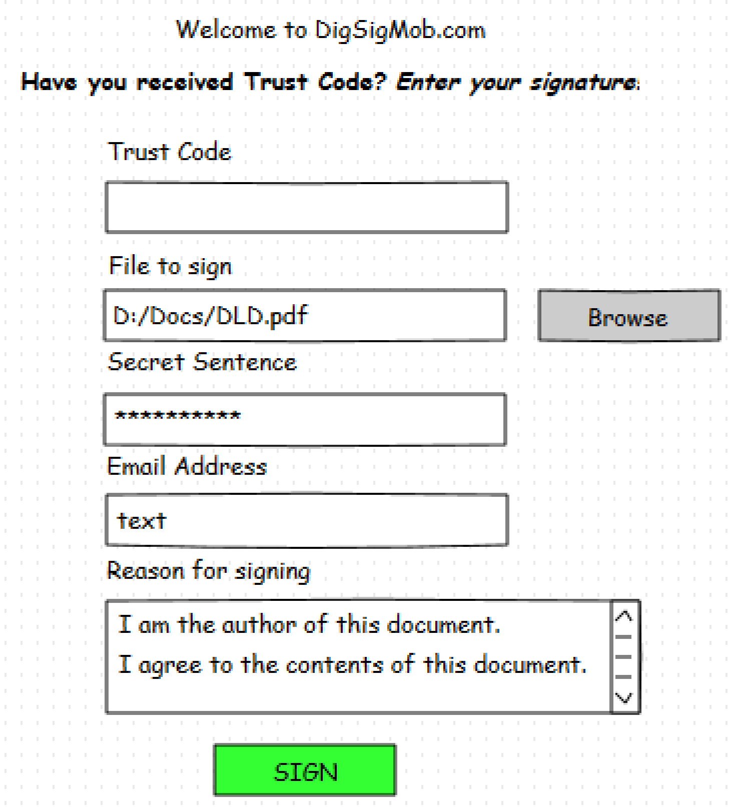
****

Email Key

Mobile key

Primary Email address

Co-signer



Welcome to DigSigMob.com

Have you received Trust Code? *Enter your signature*.

Trust Code

File to sign

D:/Docs/DLD.pdf

Secret Sentence

****

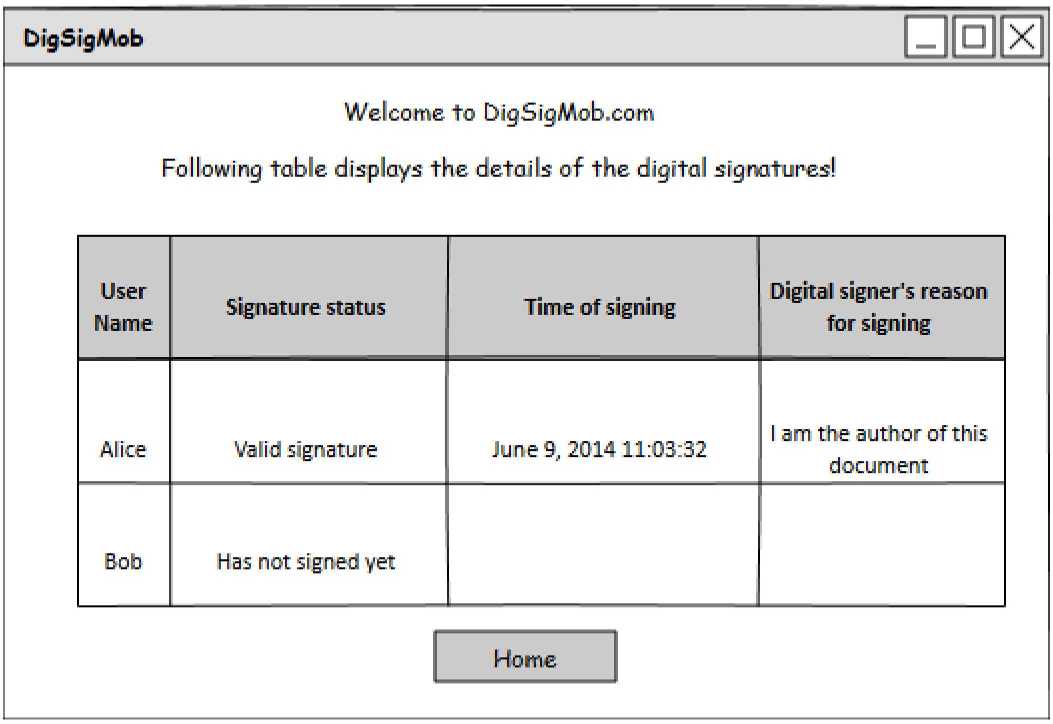
Email Address

text

Reason for signing

I am the author of this document.
I agree to the contents of this document.

Message



Welcome to DigSigMob.com

Following table displays the details of the digital signatures!

User Name	Signature status	Time of signing	Digital signer's reason for signing
Alice	Valid signature	June 9, 2014 11:03:32	I am the author of this document
Bob	Has not signed yet		

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. 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.

3. DigitalSignatureBean

Class name

Public class DigitalSignatureBean

Functionality

Bean class that defines the digital signature.

Variable names

1. private UserID userId
2. private TrustCode trustCode
3. private String reasonForSigning
4. private byte[] signedFile
5. private Timestamp signedTime
6. private boolean hasSigned
7. 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 InvalidInputException	Setter method for user id.
private boolean isValid()	Method that validates the user id.

10. RSAKeyPair

Class name

Public class RSAKeyPair

Functionality

Data type that defines the RSA key pair.

Variable names

private RSAPublicKey publicKey
private RSAPrivateKey privateKey

Constructor

public RSAKeyPair(RSAPrivateKey privateKey, RSAPublicKey publicKey)

Methods

Signature	Functionality
public RSAPublicKey getPublicKey()	Getter method for RSA public key.
public void setPublicKey(RSAPublicKey publicKey)	Setter method for RSA public key.
public RSAPrivateKey getPrivateKey()	Getter method for RSA private key.
public void setPrivateKey(RSAPrivateKey privateKey)	Setter method for RSA private key.

Model: Business classes

1. SignatureManager

Class name

Public class SignatureManager

Functionality

Business class that creates a digital signature.

Variable name

Provider BC

Constructor

public SignatureManager()

Methods

Signature	Functionality
public byte[] createSignature(PrivateKey privateKey, byte[] data)	Method that creates digital signatures.

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[] solveLinearCongruence(BigInteger a, BigInteger b, BigInteger m)	Returns a solution of x for linear congruences of the form ax 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.DigSigMobClientUtils

Class name

Public class DigSigMobServerUtils

Functionality

Util class that finds prime numbers.

Variable names

public static final BigInteger *R*

Methods

Signature	Functionality
public static BigInteger generatePrime(int bits)	method generates a prime number of the size specified by the parameter bits.

6.PrimeGenerator

Class name

Public class PrimeGenerator

Functionality

Util class that finds various types of prime numbers viz. safe and strong.

Variable names

int minBitLength
int certainty
Random sr

Methods

Signature	Functionality
public PrimeGenerator(int minBitLength, int certainty, Random random)	Method generates a prime number given the required specifications.
public BigInteger getStrongPrime()	This method finds and returns a strong prime.
public BigInteger getSafePrime()	This method returns a safe prime of form 2rt+1 where t is a large prime.
public BigInteger getNextSafePrime(BigInteger minimumValue)	This method returns the next safe prime.
public BigInteger[] getSafePrimeAndGenerator()	This method returns a safe prime of form 2rt+1 where t is a large prime and the factorization of r is known. It also returns a generator for the safe prime.

7.RSA

Class name

Public class RSA

Functionality

Util class that implements the RSA algorithm.

Variable names

public static final BigInteger R1
private static final int seed

private static final int STRENGTH

Methods

Signature	Functionality
public static double testgenerateKeys(int p1, int p2, String fingerprint, BigInteger R, PrintWriter out)	Method that generates the keys.
public static KeyPair generateKeys(String p1, String p2, String secret)	Method that generates the key pair.

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. UserRegistration

Class name

Public class UserRegistration

Functionality

Control class responsible for user registration.

Variables

private SocketManager socketManager
private UICallback ui

Constructor

public UserRegistration(UICallback ui)

Methods

Signature	Functionality
public void registerUser(UserBean user)	Method that registers the users.

2. CertificateCreation

Class name

Public class CertificateCreation

Functionality

Control class responsible for certificate creation.

Constructor

public CertificateCreation(UICallback ui)

Variables

private SocketManager socketManager
private UICallback ui
private String keyP1
private String keyP2
private String secret
private EmailAddress email

Methods

Signature	Functionality
public void createCertificate(String keyP1, String keyP2, String secretS, EmailAddress email)	Method that creates the certificate for users.

3.SignatureCreationOfInitiator

Class name

Public class SignatureCreationOfInitiator

Functionality

Class that is responsible for signature creation of initiator.

Constructor

public SignatureCreationOfInitiator(SocketManager socket, UICallback ui)

Variables

```
private SocketManager socketManager
private UICallback ui
private int status
private String reason
private DocumentBean document
private String secret
private EmailAddress[] emails
```

Methods

Signature	Functionality
public void createSignature(EmailAddress email, DocumentBean origDocBn, EmailAddress reqdCosigner, String secretS, String signingReason)	Method that creates the signature for initiators.

3.SignatureCreationOfCoSigner

Class name

Public class SignatureCreationOfCoSigner

Functionality

Class that is responsible for signature creation of co-signer.

Constructor

public SignatureCreationOfCoSigner(UICallback ui)

Variables

```
private SocketManager socketManager
private UICallback ui
private int status
private String reason
private DocumentBean document
private String secret
private EmailAddress email
```

Methods

Signature	Functionality
public void createCoSignerSignature(TrustCode trCode, EmailAddress coSigEmail, DocumentBean cosignDOc, String	Method that creates the signature for co-signer.

secretS, String reason)	

5.TrustCodeVerification

Class name

Public class TrustCodeVerification

Functionality

Control class responsible for trust code verification.

Variables

```
private SocketManager socketManager
private UICallback ui
```

Constructor

public TrustCodeVerification(UICallback ui)

Methods

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

Controller: Socket classes

1. ListenFromServer

Class name

public class ListenFromServer

Functionality

Util class for socket connextion.

Variables

```
private ObjectInputStream sInput
private ResponseListener delegate
```

Constructor

public ListenFromServer(ObjectInputStream input, ResponseListener delegate)

2. SocketManager

Class name

Public class SocketManager

Functionality

Util class for socket connection.

Variable names

```
private static SocketManager socketManager
private ObjectInputStream sInput
private ObjectInputStream sOutput
private Socket socket
private boolean isConnected
private int port
private String server
private ResponseListener respListener
private ConnectionListener connListener
private ListenFromServer listenFromServer
```

Constructor

private SocketManager(String server, int port, ResponseListener respListener, ConnectionListener connListener)

Methods

Signature	Functionality
public boolean isConnected()	Getter method for isConnected.
public static SocketManager getInstance(String server, int port, ResponseListener respListener, ConnectionListener connListener)	Method for managing socket actions.
public void setResponseListener (ResponseListener listener)	Method for managing socket actions.
public void setConnectionListener (ConnectionListener listener)	Method for managing socket actions.
public void connect()	Method for managing socket actions.
protected void disconnect()	Method for managing socket actions.
public void start()	Method for managing socket actions.
public void sendMessage(final SocketMessage msg)	Method for managing socket actions.